Week2

2017年9月11日 16:37

I Introduction

1) what's machine learning

① two problems

- classification / pattern recognition
- detection / hypothesis testing
- regression / prediction / estimation / filtering

classification

example: image recognition

input: image

output: "what is the image"

mathematically:

$x \longrightarrow [f(\cdot)] \longrightarrow c$

real vector $x \in \mathbb{R}^n$

$c \in \{c_1, c_2, \cdots c_k\} = \mathbb{C}$

a people playing frisbee

$f: \mathbb{R}^n \to \mathbb{C}$

estimation / regression

Example: price prediction / forecasting

input: price at $t, t-1, t-2 \cdots t-n$ (real vector)

Output: price at $t+k$ $(k>0)$ (real number)

Mathmatically:

$x \longrightarrow [f(\cdot)] \longrightarrow y$

$x \in \mathbb{R}^n$ $y \in \mathbb{R}^m$

$f: \mathbb{R}^n \to \mathbb{R}^m$

2017年9月11日 16:37

② Machine learning

To get $f(\cdot)$ → Design $f(\cdot)$
- requires understanding (theory)

→ By machine learning
- requires "data" training data.

D: training data
M.L. D: → $\hat{f}(\cdot)$

2) mathematical foundation

- Probability theory & statistics
- Optimization

2. Review: Prob. RV & statistics

1) Probability

① Prob. model, Prob Axioms, Prob. space

Probability model
- Experiment
- Outcome
- Event
- Probability

σ-Algebra.

Sample space / universe

Ex: Roll a dice once

$\{1, 2, 3 \cdots 6\} = S$

"get an Even" $\{2, 4, 6\}$

$P["\text{get an event \#}"] = P[\{2, 4, 6\}] = \frac{1}{2}$

3

2017年9月11日 16:37

Event:

△ An Event A is subset of S

△ Events: A, B ... → $\mathcal{F}$ Set of event. ← Sigma. ← σ-Algebra.

If A is an event, A^c must be an event — A^c — A complement

If A & B are event, So are $A \cap B$, $A \cup B$

If $A_1, A_2 \dots A_n$ are events, → So is $\bigcup_{i=1}^{\infty} A_i$

Probability: Measure, normalized.

a Measure, function, $m(A)$

input: a set

output: non-negative number

additivity: if $A \cap B = \emptyset$, $m(A \cup B) = m(A) + m(B)$.

Normalized

$0 \le P[A] \le 1$, $P[S] = 1$

Summary: Probability model Probability space

$(S, \mathcal{F}, P)$

Sample Space — set of event, σ-Algebra — Prob (measure) normalized

$P[S] = 1$

EX:

Experiment: roll a dice once

$S = \{1, 2, 3, \dots 6\}$

$\mathcal{F}$ = all subsets of S.

P, $P[\{1\}] = \dots = 1/6$

S = Same

$f_1 = \{\{2, 4, 6\}\}$

expand f_1 such that f_1 is σ-Algebra

↳ minimally

Specify a prob for this

Sum of normally distributed random variables

$$X \sim N(\mu_X, \sigma_X^2)$$
$$Y \sim N(\mu_Y, \sigma_Y^2)$$
$$Z = X + Y,$$

then

$$Z \sim N(\mu_X + \mu_Y, \sigma_X^2 + \sigma_Y^2).$$

quiz

2017年9月11日 16:37

3 偏导函数的定义

如果$z=f(x,y)$在区域D内任一点(x,y)处对x的偏导数都存在，那么这个偏导数就是x、y的函数，称为$z=f(x,y)$对x的偏导（函）数，

记作$\frac{\partial z}{\partial x}$，$\frac{\partial f}{\partial x}$ 或 $f_x'(x,y)$

$z=f(x,y)$ 对 y 的偏导数

记作$\frac{\partial z}{\partial y}$，$\frac{\partial f}{\partial y}$，或 $f_y'(x,y)$

注 1 如何计算偏导数

在计算 $\frac{\partial z}{\partial x}$ 将y看成常量，仅对x求导

在计算 $\frac{\partial z}{\partial y}$ 将x看成常量，仅对y求导

注 2 一点的偏导数与偏导函数的关系

$\left.\frac{\partial z}{\partial x}\right|_{\substack{x=x_0\\y=y_0}}=\frac{\partial z}{\partial x}$ 在(x_0,y_0)点的函数值

https://wenku.baidu.com/view/e4e6659e51e79b8968022c9d.html?re=view

A set of subsets of Ω, $\mathcal{F}$, is called a **σ-field (σ-algebra)** if it satisfies the following 3 properties:

1. $\Omega \subset \mathcal{F}$
2. $\mathcal{F}$ is closed under complementation: If $A \in \mathcal{F}$, then so is its complement, $\Omega \setminus A = A^c \in \mathcal{F}$.
3. $\mathcal{F}$ is closed under countable unions: If $A, B \in \mathcal{F}$, then so is $A \cup B \in \mathcal{F}$.

→ This means that $A \cap B = (A^c \cup B^c)^c$ from de Morgan's law = closed under intersections.

or $(A \cup B)^c = (A^c \cap B^c)$

An algebra is a collection of subsets which is closed under some countable set operations i.e. if a set X can be obtained from subsets in $\mathcal{F}$ through complementation, union or intersection then X is also a member of $\mathcal{F}$.

The power set is the largest σ-algebra of Ω.

Trivial set ≤ σ-algebra ≤ P(Ω)

Sigma Field / sigma algebra

根据组合数和二项式定理

子集个数：Cn0+Cn1+Cn2+...+Cnn = (1+1)^n=2^n

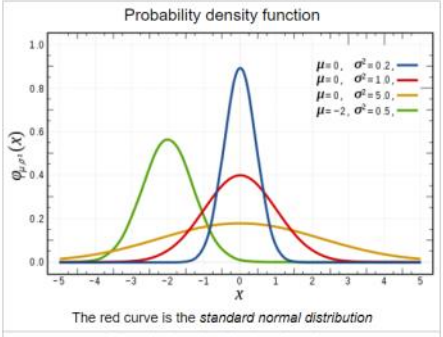
https://www.math.lsu.edu/~sengupta/7312s02/sigmaalg.p

normal (or **Gaussian**) **distribution**

Notation	$\mathcal{N}(\mu, \sigma^2)$
Parameters	$\mu \in \mathbb{R}$ — mean (location) $\sigma^2 > 0$ — variance (squared scale)
Support	$x \in \mathbb{R}$
PDF	$\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
CDF	$\frac{1}{2}\left[1+\operatorname{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right]$
Quantile	$\mu+\sigma\sqrt{2}\operatorname{erf}^{-1}(2F-1)$
Mean	μ
Median	μ
Mode	μ
Variance	σ^2
Skewness	0
Ex. kurtosis	0
Entropy	$\frac{1}{2}\log(2\pi e\sigma^2)$

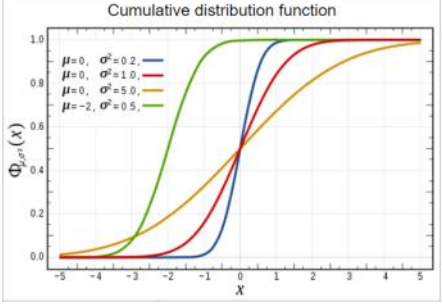
Normal distribution

Probability density function



The red curve is the *standard normal distribution*

Cumulative distribution function



Skewness	0
Ex. kurtosis	0
Entropy	$\frac{1}{2}\log(2\pi e\sigma^2)$
MGF	$\exp\{\mu t+\frac{1}{2}\sigma^2t^2\}$
CF	$\exp\{i\mu t-\frac{1}{2}\sigma^2t^2\}$
Fisher information	$\begin{pmatrix}1/\sigma^2 & 0\\ 0 & 1/(2\sigma^4)\end{pmatrix}$

https://www.zhihu.com/question/26055805

Week3

2017年9月18日 16:37

1) Joint & Conditional probs.

Joint: $P[A \cap B]$ ← joint

Conditional $P[A|B] = \frac{P(A \cap B)}{P(B)}$

⇒ Bayesian influence

$P[A \cap B] = P[A|B] \cdot P[B]$

? independence

if $P[A|B] = P[A]$

$P[A \cap B] = P[A] \cdot P[B]$

- Reverse influence. (obs. / cause)

$P[B|A] = \frac{P[A|B] \cdot P[B]}{P[A]}$ (cause / obs.)

- Total prob formula.

$P[A] = P[A \cap B] + P[A \cap B^c]$

$= P[A|B] \cdot P[B] + P[A|B^c] \cdot P[B^c]$

2) Random Variables.

A random variable:

① basic idea: convert (S, f, P) into $(\mathbb{R}, B, P)$ → pdf, $P(\text{Event}) = \int \text{pdf}$.

($\mathbb{R}$: real number space; B: event generate by interval)

Example: flip a coin once.

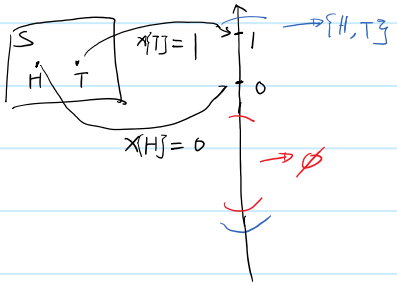
$(S, f, P) \longrightarrow (\mathbb{R}, B, P)$

$n=2$ $S = \{H, T\}$

2^n $f = \{\{H\}, \{T\}, \{H, T\}, \emptyset\}$

$P[\{H\}] = 1/2$

X is a measureable function corr. to f.



$X^{-1}(A) = \{w \in S \mid X(w) \in A\}$

$X^{-1}(A) \in f$

inverse image; interval

② random variable:

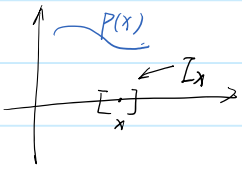
measurable mapping from S to $\mathbb{R}$

2017年9月18日 16:37

3) Pdf. (Probability density function)

X: random variable. eg.

$$P_X(x) = P(x) = \lim_{|I_x| \to 0} \frac{P[X \in I_x]}{|I_x|}$$



$$P[\text{event}] = P[X \in A] = \int_A P(x)\,dx$$

Normal

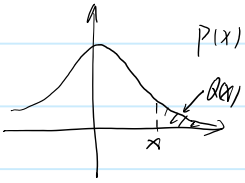
4) Gaussian r.v.

$$X: P(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-u)^2/2\sigma^2}$$ (mean u, variance σ^2)

$X \sim N(u, \sigma^2)$

$N(m, \sigma^2)$

Q-function: tails of a Gaussian.



$$P(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \quad x \in (0,1)$$

standard

$Q(0) = 1/2$

$Q(+\infty) = 0$

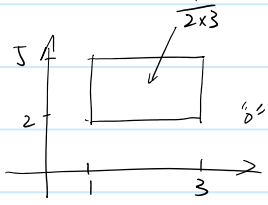
$$\int_x^{+\infty} P(x)\,dx = P[X \geq x]$$

5) Joint & conditional Pdfs.

joint: pdf of X and Y. e.g. (X: length., Y: width)

$P(x,y)$

$$P[\text{event related to } x \text{ & } y] = P[(x,y) \in A] = \iint_A P(x,y)\,dx\,dy$$



P[length > 2, width < 3]

$$= \int_2^3 \int_2^3 \frac{1}{6}\,dx\,dy$$

$$= \int_2^3 dx \int_2^3 dy\, \frac{1}{6} = \frac{1}{6}$$

3

2017年9月18日 16:37

Conditional 2

$P(x|y)$ = pdf of X given $Y=y$.

$$= \frac{P(x,y)}{P(y)}$$ ← marginal

$P(y) = \int_{-\infty}^{+\infty} P(x,y)\,dx$

Bayesian influence. likelihood prior.

$$P(x|y) = \frac{P(y|x)\,P(x)}{P(y)}$$ reverse influence.

$P(y) = \int_{-\infty}^{+\infty} dx\, P(y|x)\,P(x)$.

cause or data not observable. obs. data. normalization factor.

Example: ☆ Multivariate Gaussian. Jointly Gaussian pdf.

$$X = (X_1, X_2, X_3 \cdots X_n)^T = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \vec{x}$$

$$P(X) = P(X_1, X_2 \cdots X_n) = (2\pi)^{-\frac{n}{2}} |\Sigma|^{-\frac{1}{2}}$$

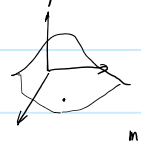
$|\Sigma|$: covariance matrix; determinant.

$$e^{-\frac{1}{2}(X-M)^T \Sigma^{-1} (X-M)}$$

M: mean vector

$[\ ]_{1\times n}\ [\ ]_{n\times n}\ [\ ]_{n\times 1}$.

Example $n=2$.



$M = \begin{bmatrix} m_1 \\ m_2 \end{bmatrix}$

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{n_1} \\ \hline X_{n_1+1} \\ \vdots \\ X_n \end{bmatrix} \begin{matrix} \} X_1 \\ \text{partition} \\ \} X_2 \end{matrix}$$

$$M = \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} \begin{matrix} \leftarrow n_1 \text{ dim} \\ \leftarrow n-n_1 \text{ dim} \end{matrix}$$

uncertainty about X ⇐

$$\Sigma \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ n_1\times n_1 & n_1\times(n-n_1) \\ \hline \Sigma_{21} & \Sigma_{22} \\ (n-n_1)\cdot n_1 & (n-n_1)\cdot(n-n_1) \end{bmatrix}$$

$$P(X_1|X_2) = \frac{P(X_1, X_2)}{P(X_2)} = \frac{P(X)}{P(X_2)}$$

$$P(X_2) = \int dX_1\, P(X_2|X_1)\,P(X_1)$$

n_1 fold integration.

正定 Positive Definite

$$\begin{bmatrix} x^T \end{bmatrix} \begin{bmatrix} \Sigma \end{bmatrix} \begin{bmatrix} x \end{bmatrix} > 0$$

Conclusion: X_1 given X_2 is Gaussian

$$P(X_1|X_2) \sim N(m_{1|2}, \Sigma_{1|2})$$

$m_{1|2}$: conditional mean; $\Sigma_{1|2}$: conditional covariance matrix.

$$P(X_1|X_2) = 2\pi^{-\frac{n_1}{2}} |\Sigma_{1|2}|^{-\frac{1}{2}} \cdot e^{-\frac{1}{2}(x-m_{1|2})^T \Sigma_{1|2}^{-1} (X_1 - m_{1|2})}$$

$$m_{1|2} = m_1 + \Sigma_{12}\Sigma_{22}^{-1}(X_2 - m_2)$$

$$\Sigma_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$$

revised term after condition.

non-negative

4

2017年9月18日 16:37

measurable space

A **measurable space** is a set with a distinguished σ-algebra of subsets (called measurable). More formally, it is a pair $(X, \mathcal{A})$ consisting of a set X and a σ-algebra $\mathcal{A}$ of subsets of X.

Let $(X, \mathcal{A})$ and $(Y, \mathcal{B})$ be measurable spaces.

- A map $f : X \to Y$ is called *measurable* if $f^{-1}(B) \in \mathcal{A}$ for every $B \in \mathcal{B}$.
- These two measurable spaces are called *isomorphic* if there exists a bijection $f : X \to Y$ such that f and f^{-1} are measurable (such f is called an isomorphism).

Let X be a set, $(Y, \mathcal{B})$ a measurable space, and $(f_i)_{i \in I}$ a family of maps $f_i : X \to Y$. The σ-algebra *generated* by these maps is defined as the smallest σ-algebra $\mathcal{A}$ on X such that all f_i are measurable from $(X, \mathcal{A})$ to $(Y, \mathcal{B})$. More generally, one may take measurable spaces $(Y_i, \mathcal{B}_i)$ and maps $f_i : X \to Y_i$. On the other hand, if Y is $\mathbb{R}$ (or $\mathbb{C}$, $\mathbb{R}^n$ etc.) then $\mathcal{B}$ is by default the Borel σ-algebra.

From <https://www.encyclopediaofmath.org/index.php/Measurable_space>

notation:

1. X(ω) to denote the **numerical value** of a random variable X
2. the **probability** that the outcome of the experiment is such that X is no larger than some c,

i.e., that the **outcome** belongs to the **set** {ω | X(ω) ≤ c}.

in order to have a probability assigned to that set, we need to make sure that it is **F-measurable**.

Definition 1. *(**Random variables**) Let $(\Omega, \mathcal{F})$ be a measurable space.*

(a) A function $X : \Omega \to \mathbb{R}$ is a **random variable** *if the set $\{\omega \mid X(\omega) \leq c\}$ is $\mathcal{F}$-measurable for every $c \in \mathbb{R}$.*

(b) A function $X : \Omega \to \overline{\mathbb{R}}$ is an **extended-valued random variable** *if the set $\{\omega \mid X(\omega) \leq c\}$ is $\mathcal{F}$-measurable for every $c \in \overline{\mathbb{R}}$.*

We note here a convention that will be followed throughout: we will always use upper case letters to denote random variables and lower case letters to denote numerical values (elements of $\overline{\mathbb{R}}$). Thus, a statement such as "$X(\omega) = x = 5$" means that when the outcome happens to be ω, then the realized value of the random variable is a particular number x, equal to 5.

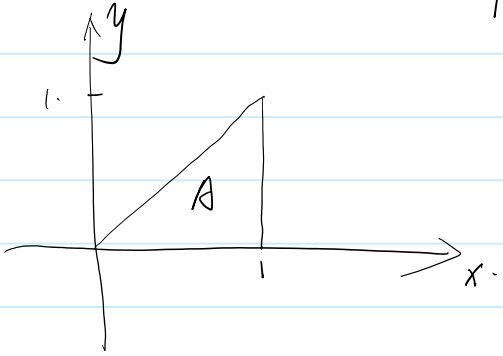
1.2 The law of a random variable

For a random variable X, the event $\{\omega \mid X(\omega) \leq c\}$ is often written as $\{X \leq c\}$, and is sometimes just called "the event that $X \leq c$." The probability of this event is well defined, since this event belongs to $\mathcal{F}$. Let now B be a more general subset of the real line. We use the notation $X^{-1}(B)$ or $\{X \in B\}$ to denote the set $\{\omega \mid X(\omega) \in B\}$.

Week4

2017年9月25日 16:37

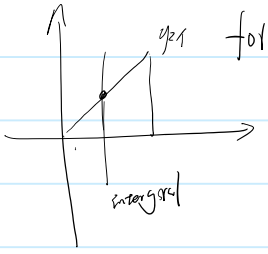
Review: Conditional pdf.



$$P(x,y)=\begin{cases} c\cdot y, & (x,y)\in A \\ 0, & (x,y)\notin A \end{cases} \qquad c>0$$

find $P(y|x)=?$

$$P(y|x)=\frac{P(x,y)}{P(x)}=\frac{P(x\in(0,1),\ y\in(0,1))}{P(x)}$$



for $x\in(0,1)$

$$P(x)=\int_{-\infty}^{+\infty}P(x,y)\,dy$$

$$P(x)=\int_0^x dy\, cy=\frac{c}{2}x^2.$$

$$=\frac{cy}{\frac{c}{2}x^2}$$

$$=\frac{2y}{x^2} \qquad (x,y)\in A$$

$$=0 \qquad (x,y)\notin A$$

Random Gaussian Vector. n-dim.

$$X=\begin{bmatrix} X_1 \\ \hline X_2 \end{bmatrix} \quad \begin{matrix} n_1\,dim \\ n_2\,dim \end{matrix} \qquad n_1+n_2=n$$

$$\Rightarrow m=\begin{bmatrix} m_1 \\ \hline m_2 \end{bmatrix} \quad \begin{matrix} n_1\,dim \\ n_2\,dim. \end{matrix}$$

$X\sim N(m,\Sigma)$ ← uncertainty related x.

$$\left[\begin{array}{c|c} \Sigma_{11} & \Sigma_{12} \\ \hline \Sigma_{21} & \Sigma_{22} \end{array}\right]$$

Σ_{11}: $n_1\times n_1$, Σ_{12}: $n_1\times n_2$, Σ_{21}: $n_2\times n_1$, Σ_{22}: $n_2\times n_2$.

Conditional mean

$$m_{1/2}=m_1+\Sigma_{12}\Sigma_{22}^{-1}(X_2-m_2)$$

(Σ_{22}^{-1}: (uncertainty of x_2)$^{-1}$)

$$\Sigma_{1/2}=\Sigma_{11}-\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$$

$\Sigma_{1/2}$: $Cov[X_1|X_2]$

Σ_{12}: Correlation between X_1 & X_2

$P(X_1|X_2)$: Gaussian $\sim N(m_{1/2},\Sigma_{1/2})$

$m_{1/2}$: n_1 dim; $\Sigma_{1/2}$: $n_1\times n_1$

remaining uncertainty given X_2.

2

2017年9月25日 16:37

Example

$$X = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \begin{matrix} x_1 \\ x_2 \end{matrix} \sim N(m, \Sigma)$$

$$m = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \begin{matrix} m_1 \\ m_2 \end{matrix}$$

$$\Sigma = \left[\begin{array}{c|cc} 1 & \frac{1}{2} & \frac{1}{3} \\ \hline \frac{1}{2} & 2 & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & 3 \end{array}\right]$$

Σ_{11} Σ_{12} Σ_{21} Σ_{22}

2. Symmetric

smaller $\Rightarrow$ positive definite

$$m_{1|2} = m_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - m_2)$$

$$= 0 + \left[\frac{1}{2}, \frac{1}{3}\right] \cdot \begin{bmatrix} 2 & \frac{1}{4} \\ \frac{1}{4} & 3 \end{bmatrix}^{-1} \cdot \begin{bmatrix} u_2 \\ u_3 \end{bmatrix}$$

1×2 2×2 1×2

$$[a, b] \cdot \begin{bmatrix} u_2 \\ u_3 \end{bmatrix}$$

$$= au_2 + bu_3$$

1×1

1×2 2×2 2×1

$$\Sigma_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} = 1 - \left[\frac{1}{2}, \frac{1}{3}\right] \cdot \begin{bmatrix} 2 & \frac{1}{4} \\ \frac{1}{4} & 3 \end{bmatrix}^{-1} \times \begin{bmatrix} \frac{1}{2} \\ \frac{1}{3} \end{bmatrix}$$

---

3) Statistics of random variables

① Expectations (mean)

X : r.v. $P(x)$

$$E[X] = \int_{-\infty}^{+\infty} x P(x)\, dx$$

$$E[g(x)] = \int_{-\infty}^{+\infty} g(x) P(x)\, dx$$

$$E\left[(X - E[X])^2\right] = \text{Variance of } X.$$

$$E[g(x,y)] = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} g(x,y) P(x,y)\, dx\, dy$$

Special case.

$E[XY]$: correlation.

$E[(X - E[X])(Y - E[Y])] = \text{Cov}[X, Y]$

2017年9月25日 16:37

2. Classical theory of classification & estimation.

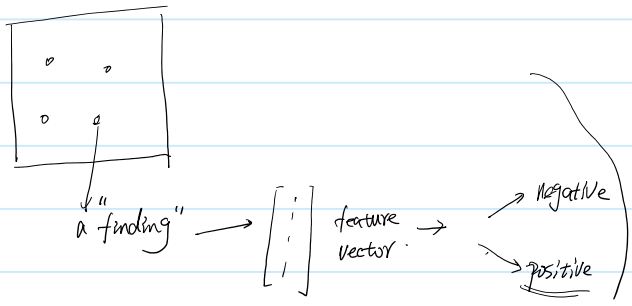
1) why? what?

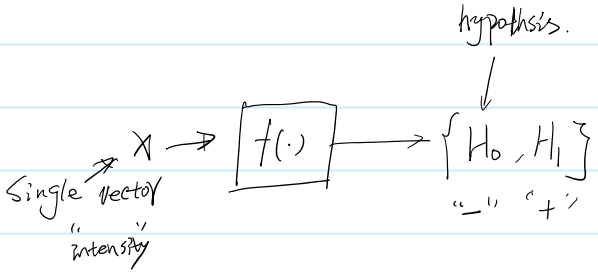
machine learning
- is estimate. Data → $f(\cdot)$
  learning what $f(\cdot)$ is.
- (detection) classification → $x \to \boxed{f(\cdot)} \to \{C_1, C_2, \cdots C_n\}$
- estimation. (regression) → $x \to \boxed{f(\cdot)} \to y \in \mathbb{R}^m$.

classical theory: what's the best $f(\cdot)$.
(with ideal assumption)

2) classical detection theory.

① Simple example: tumor detection





Question: what's the best possible $f(\cdot)$.

2017年9月25日 16:37

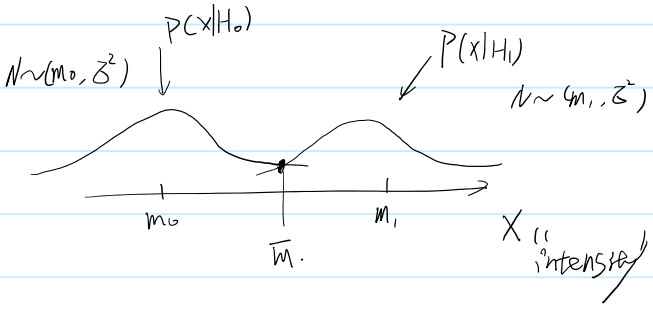
② modeling:

- $P[H_0] = P_0 \quad P[H_1] = P_1$

  $P_0 + P_1 = 1.$

- $P(x|H_0)$ = conditional pdf given H_0

  $P(x|H_1)$ = conditional pdf given H_1



③ optimal $f(\cdot)$: (classifier)

Quality measure: **prob of error P_e**

Calculate P_e

$P_e = P_{e0} \cdot P_0 + P_{e1} \cdot P_1$

$= P(e,0) + P(e,1)$

Given H_0, $P_{e|0} = 0$

Given H_1, $P_{e|1} = 1$

Consider 3 "doctors"

doctor #1. $f(x) = H_0$ optimist. ("negative")

doctor #2. $f(x) = H_1$ pessimist ("positive")

doctor #3. $f(x) = \begin{cases} H_0, & \text{if } x < \bar{m}. \\ H_1, & \text{if } x \geq \bar{m}. \end{cases}$ smart

P_e for each doctor.

#1. $P_{e|0} = P[\text{error} \mid x \text{ comes from } H_0] = 0$ (H_0 is true)

$P_{e|0} = P[\text{error} \mid x \text{ comes from } H_1] = 1.$

$P_e = 0 \times P_0 + 1 \times P_1 = P_1$

$P_e = 1 \times P_0 + 0 \times P_1 = P_0$

$$P_e = P_0 \times \int_{\bar{m}}^{+\infty} P(x|H_0)\,dx + P_1 \times \int_{-\infty}^{\bar{m}} P(x|H_1)\,dx$$

$$P_e = Q\left(\frac{\Delta m}{2\sigma}\right) \rightarrow \text{SNR}$$

$\bar{m} - m_0 = \frac{1}{2}(m_1 + m_0) - m_0$

$= \frac{1}{2}(m_1 - m_0)$

Δm.

$$Q(x) = \int_x^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}\,du$$

(tail function).

Summary:

⇒ None are optimal

min P_e for all P_1, $P(x|H_0)$, $P(x|H_1)$

Week5

2017年10月2日 16:37

Review: classical theory of classification
Detection
Recognition

x → [$f(\cdot)$] → $y \in \{C_1, C_2, C_3 \cdots C_k\}$

input

Categories.

Special case (Binary)

x → [$f(\cdot)$] → $y \in \{H_0, H_1\}$

Machine learning:

D: $\{(x, y)\}$ = training data.

$D \rightarrow f(\cdot)$ good classifier

Classical Theory:

model of x & y → best possible $f(\cdot)$

$P(H_0) = P_0$ $(P_1 = 1 - P_0)$

$P(x|H_0)$, $P(x|H_1)$

$\uparrow y = H_0$ $\uparrow y = H_1$

model

→ $f(\cdot)$ best "in some way" ($\min P_e$)

2017年10月2日 16:37

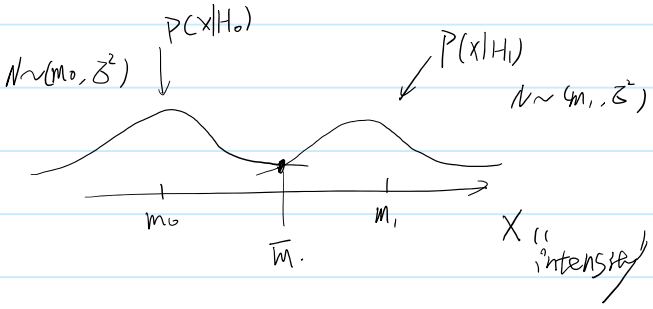
② modeling:

- $P[H_0] = P_0 \quad P[H_1] = P_1$

  $P_0 + P_1 = 1.$

- $P(x|H_0)$ = conditional pdf given H_0

  $P(x|H_1)$ = conditional pdf given H_1



③ optimal $f(\cdot)$: (classifier)

Quality measure: **prob of error P_e**

Calculate P_e

$P_e = P_{e0} \cdot P_0 + P_{e1} \cdot P_1$

$= P(e,0) + P(e,1)$

Given H_0, $P_{e|0} = 0$

Given H_1, $P_{e|1} = 1$

Consider 3 "doctors"

doctor #1. $f(x) = H_0$ optimist. ("negative")

doctor #2. $f(x) = H_1$ pessimist ("positive")

doctor #3. $f(x) = \begin{cases} H_0, & \text{if } x < \bar{m} \\ H_1, & \text{if } x \geq \bar{m} \end{cases}$ smart

P_e for each doctor:

#1. $P_{e|0} = P[\text{error} \mid x \text{ comes from } H_0 \text{ is true}] = 0$

$P_{e|1} = P[\text{error} \mid x \text{ comes from } H_1] = 1.$

$P_e = 0 \times P_0 + 1 \times P_1 = P_1$

$P_e = 1 \times P_0 + 0 \times P_1 = P_0$

$$P_e = P_0 \times \int_{\bar{m}}^{+\infty} P(x|H_0)\,dx + P_1 \times \int_{-\infty}^{\bar{m}} P(x|H_1)\,dx$$

$$P_e = Q\left(\frac{\Delta m}{2\sigma}\right) \rightarrow \sqrt{SNR}$$

$\bar{m} - m_0 = \frac{1}{2}(m_1 + m_0) - m_0$

$= \frac{1}{2}(m_1 - m_0)$

Δm

$$Q(x) = \int_x^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}\,du$$

(tail function)

Summary:

⇒ None are optimal

min P_e for all P_1, $P(x|H_0)$, $P(x|H_1)$

2017年10月2日 16:37

Optimal classifier derivation..

① Any $f(\cdot) \equiv (R_0, R_1)$ $\quad R_1 \cup R_0 = \mathbb{R}$

② $P_e = P_{e|0} P_0 + P_{e|1} P_1$

$P_{e|0} \uparrow P[x \in R_1 | H_0]$

$\Rightarrow f^*(\cdot)$ (optimal $f(\cdot)$)

$$P_e = P[x \in R_1 | H_0] \cdot P_0 + P[x \in R_0 | H_1] \cdot P_1$$

$$= P_0 \int_{R_1} P(x|H_0) + P_1 \int_{R_0} P(x|H_1)$$

$$= P_0 \int_{R_1} P(x|H_0) + P_1 \left[1 - \int_{R_1} P(x|H_1)\right]$$

$$= P_1 \int_{R_1} \left(P_0 \cdot P(x|H_0) - P_1 \cdot P(x|H_1)\right)$$

$= \underline{P_e(R_1)}$ - a function of R_1

R_1^* + $P_1 P(x|H_1) > P_0 P(x|H_0)$ } ⇐ a set of R_1 (interval / range)

$\Rightarrow P_e$ minimized.

Subset of real line.

$R_0^* = (R_1^*)^c$.

optimal classifier. $\Rightarrow$ leads to R_1

$$P_1 P(x|H_1) \underset{H_0}{\overset{H_1}{\gtrless}} P_0 P(x|H_0).$$

likehood ratio LRT $\quad \dfrac{P(x|H_1)}{P(x|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \dfrac{P_0}{P_1}$

optimal classifier / Bayesian classifier

OR

$$\log\left(\frac{P(x|H_1)}{P(x|H_0)}\right) \underset{H_0}{\overset{H_1}{\gtrless}} \log\left(\frac{P_0}{P_1}\right)$$

2017年10月2日 16:37

Example P_0 ($P_1 = 1 - P_0$). $P(x|H_0) \sim N(m_0, \sigma^2)$, $P(x|H_1) \sim N(m_1, \sigma^2)$

Optimal classifier.

$$\log \frac{P(x|H_1)}{P(x|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \frac{P_0}{P_1}$$

$$\log \frac{\frac{1}{\sqrt{2\pi}\sigma} e^{-(x-m_1)^2/2\sigma^2}}{\frac{1}{\sqrt{2\pi}\sigma} e^{-(x-m_0)^2/2\sigma^2}}$$

$$= -\frac{(x-m_1)^2}{2\sigma^2} + \frac{(x-m_0)^2}{2\sigma^2}$$

$$= \frac{1}{2\sigma^2}\left[(x-m_0)^2 - (x-m_1)^2\right] = \frac{1}{2\sigma^2}\left[2\left(x - \frac{(m_0+m_1)}{2}\right)(m_0-m_1)\right]$$

$$= \frac{1}{\sigma^2}(x-\bar{m})\Delta m.$$

$$\Rightarrow \frac{1}{\sigma^2}(x-\bar{m})\Delta m \underset{H_0}{\overset{H_1}{\gtrless}} \log\left(\frac{P_0}{P_1}\right)$$

$$(x-\bar{m}) \underset{H_0}{\overset{H_1}{\gtrless}} \frac{\sigma^2}{\Delta m}\log\left(\frac{P_0}{P_1}\right)$$

$$x \underset{H_0}{\overset{H_1}{\gtrless}} \frac{\sigma^2}{\Delta m}\log\left(\frac{P_0}{P_1}\right) + \bar{m}$$

$\Longrightarrow$ $P_0 = P_1$ or Δm is very large

Compare:

doctor #1: $x \underset{H_0}{\overset{H_1}{\gtrless}} +\infty$

doctor #2: $x \underset{H_0}{\overset{H_1}{\gtrless}} -\infty$

doctor #3: $x \underset{H_0}{\overset{H_1}{\gtrless}} \bar{m}$.

Extension to more than two classes.

model $P_k = P[H_k]$ $k = 0, 1, 2, \cdots K-1$ $(K>2)$

$P(x|H_k)$

Optimal classifier. assign x to H_{k^*}

$$\text{if } k^* = \underset{0\le k\le K-1}{\arg\max}\, P[H_k|x] = \underset{0\le k\le K-1}{\arg\max} \frac{P(x|H_k)P_k}{P(x)} = \underset{0\le k\le K-1}{\arg\max}\left(\log P(x|H_k) + \log P_k\right)$$

$$x^* = \underset{x}{\arg\min}\,(x-1)^2$$

$x^* = 1$, $(x^*-1)^2 = 0$

5

2017年10月2日 16:37

Extention to multiple dimensions

$P(x|H_k) \longrightarrow P(\vec{x}|H_k)$

x: 1-dim, $\vec{x}$: n-dim

Example:

H_0: $X \sim N(m_0, \Sigma_0)$

H_1: $X \sim N(m_1, \Sigma_1)$

$m_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ $m_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$\Sigma_0 = I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ $\Sigma_1 = I$

find: optimal classifier, when $P_0 = P_1 = \frac{1}{2}$.

Region: R_0 and R_1

n-dim Gaussian $N(m, \Sigma)$

$P(\vec{x}) = (2\pi)^{-\frac{n}{2}} |\Sigma|^{-\frac{1}{2}} e^{-\frac{1}{2}(\vec{x}-m)^T \Sigma^{-1} (\vec{x}-m)}$

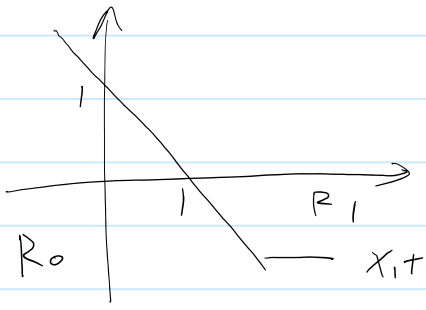
$m \to n_1$

$n \to m_0$

$$\log \frac{P(\vec{x}|H_1)}{P(\vec{x}|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \log \frac{P_0}{P_1}$$

$\log \frac{P_0}{P_1} = 0$

ans: $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, $x_1 + x_2 - 1 \underset{H_0}{\overset{H_1}{\gtrless}} 0$



Decision Boundary

Week6

2017年10月9日 16:37

Review:

Optimal (Bayesian classifier)

X: input

y: output, $y \in \{H_0, H_1\}$

models: $P_0 = P[H_0]$ $P_1 = P[H_1]$

$P(X|H_0)$, $P(X|H_1)$

Optimal classifer: LRT

$$\log \frac{P(X|H_1)}{P(X|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \log \frac{P_0}{P_1}$$

threashhold.

General LRT

$$\log \frac{P(X|H_1)}{P(X|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \lambda$$

$$\Downarrow$$

$$\log \lambda.$$

when $\lambda = \frac{P_0}{P_1}$, Bayesian classifier

$P_{e|0}$ = Prob of false alarm = P_{FA}

$P_{e|1}$ = Prob of miss = P_m.

$P_D = 1 - P_m$

Prob of detection.

detection theory

$$\begin{cases} P_D = P_D(\lambda) \\ P_{FA} = P_{FA}(\lambda) \end{cases} \Rightarrow \text{ROC curve. } (P_{FA}(\lambda), P_D(\lambda))$$

Receiver operating characteristics.

Example:

$P(X|H_0) \sim N(-1, 1)$

$P(X|H_1) \sim N(1, 1)$

$$\log \frac{P(X|H_1)}{P(X|H_0)} \gtrless \lambda.$$

assume $P_0 = P_1 = \frac{1}{2}$ $P_D = P_D(\lambda) = \cdots$

find P_D VS P_{FA} as a function of λ. $P_{FA} = P_{FA}(\lambda) = \cdots$

2

2017年10月9日 16:37

Practical problems with Optimal classifier

Traing data: $D = \{(x_i, y_i)\}_{i=1}^{n}$

x_i: real from $\mathbb{R}^m$.

y_i: H_0 or H_1

Don't have: $P(X|H_0)$, $P(X|H_1)$.

Solution:

① density estimation. (estimate)

$D \rightarrow \hat{P}(X|H_0)$, $\hat{P}(X|H_1)$, ...

Dis: requires simple model, ex: Gaussian.

no work well, with high-D data: ex: non-Gaussian.

② $D \rightarrow$ classification function $f(\cdot)$

3) Classical Estimation Theory

① Estimation problems:

Density estimation

Variable / function estimation

Density estimation:

$D = \{x_1, x_2, \cdots x_n\}$

= i.i.d. Samples of X

independent, identically distributed.

$D \longrightarrow \hat{P}(x)$

3

2017年10月9日 16:37

Density estimation:

- Parametric: $p(x) = p(x|\theta)$ e.g. Gaussian $\theta \sim (m, \sigma^2)$
  - Non-Parametric
- e.g. MLE
  - frequentist: no assumptions on $p(x)$ "Complete data driven"
  - Bayesian: assumes some prior on $p(x)$ e.g. $p(\theta)$ MAP

Variable / function estimation.

$D = \{(x_1, y_1), (x_2, y_2), \cdots (x_n, y_n)\}$

= random, i.i.d. samples of X and Y. (X, Y).

$D \rightarrow \hat{Y} = f(X)$

---

② Parametric density estimation.
(parameters estimation)

frequentist: MLE (max likelihood estimation)

Bayesian: MAP (max a posteriori)

2017年10月9日 16:37

Example: Estimating a mean of a Guassian.

X_1 X_2

$D = \{x_1, x_2 \cdots x_n\}$ = i.i.d. samples of X

$X \sim N(m, \sigma^2)$

σ^2: known.

m: unknown. to be estimated from D

$P(x|\theta) = P(x|m)$ $\theta = m$.

MLE:

$\hat{\theta}_{MLE} = \arg\max_{\theta} P(x_1, \cdots x_n|\theta)$ ∵ always exponential

$= \arg\max_{\theta} \log P(x_1 \cdots x_n|\theta)$ take log

How good is $\hat{m}$.

→ $E[\hat{m}] = ?$ unbiased. $= m$.

→ $E[(\hat{m}-m)^2] = MSE = Var[\hat{m}]$

true unknown parameter.

$E[\hat{m}] = E[\frac{1}{n}\sum x_i]$ ← random variables.

$= \frac{1}{n}\sum E[x_i]$

$= \frac{1}{n}\sum m = \frac{nm}{n} = m$

$(a+b+c\cdots z)^2 = a^2+b^2 \cdots z^2$

$= 2ab + \cdots$ C_{26}^2

$E[(\hat{m}-m)] = E[(\frac{1}{n}\sum x_i - m)^2]$

$= E[(\frac{1}{n}\sum x_i)^2 - 2m\frac{1}{n}\sum x_i + m^2]$

$= -2\hat{m}m + m^2$.

$=$

5

2017年10月9日 16:37

Q function and Error functions

2017年10月9日 16:37

https://www.gaussianwaves.com/2012/07/q-function-and-error-functions/

Week7

2017年10月16日 16:37

Review: MLE

Estimation
- Density estimation: $D \rightarrow \hat{P}(x)$
  - parametric
    - frequentist (MLE)
    - bayesian
  - non-parametric
- function estimation:

MLE under certain condition become the real θ when $|D| \rightarrow \infty$

$$E\left[(\hat{\theta}(n) - \theta)^2\right] \xrightarrow{n \to +\infty} 0$$

(n: sample size)

MLE:

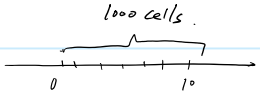
$$P(x) = P(x|\theta)$$

$$\hat{\theta} = \arg\max_{\theta} P(D|\theta)$$

$$= \arg\max_{\theta} \log P(D|\theta)$$

X: 1 dim.

100,000 points.



enough data

X: 1000 dim.

100,000 points

cells: 1000^{1000}

not enough data.

Bayesian estimation:

Can be useful when: don't have enough data.

- MAP (max a posterior)
- Conditional mean

$P(x) = P(x|\theta)$: liklihood function.

$P(\theta)$: prior density for θ.

"knowledge / constraint" about θ

MAP:

$$\hat{\theta}_{map} = \arg\max_{\theta} P(\theta|D) = \arg\max_{\theta} P(D|\theta) P(\theta) / P(D)$$

2

2017年10月16日 16:37

$$\text{MAP}: \hat{\theta}_{map} = \arg\max_{\theta} \{ P(D|\theta)\, P(\theta) \}$$

$$= \arg\max_{\theta} \{ \log P(D|\theta) + \log P(\theta) \}$$

→ If $\log P(\theta) = \text{Constant}$.

then MAP ⇒ MLE

means "no knowledge about θ"

Example: Gaussian Case

$D = \{x_1, x_2, \cdots, x_n\}$

= i.i.d. samples of X

$X \sim N(m, \sigma^2)$ (m: unknown, σ^2: known)

$\theta = m$

$P(\theta) = P(m) \sim N(m_0, \sigma_0^2)$ (assumed to be known.)

$\hat{\theta}_{map} = ?$

---

③ Variable (function estimation)

Recall: the problem.

$D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$

= i.i.d. samples of X and Y

estimation: $D \longrightarrow \hat{f}(\,)$ $\quad \hat{Y} = \hat{f}(X)$

$\hat{f}(\,)$: estimation of $f(\,) = Y = f(X)$

3

2017年10月16日 16:37

Theoretical result:

$\hat{Y} = g(X)$ any estimate

optimal estimate in sense of MMSE.

$g^*(X) = E[Y|X]$

$\hat{Y} = E[Y|X]$ is the optimal estimator of Y from X.

Week8

2017年10月23日 16:37

Review: Variable / function estimation

$D = \{(x_1, y_1), (x_2, y_2), \cdots (x_n, y_n)\}$

$=$ i.i.d. subsets from (X, Y)

estimation: $D \longrightarrow f(\cdot) : \hat{Y} = f(X) \approx Y$

~~base~~ best estimate

$\hat{Y} = f(X) = E[Y|X]$

= best estimate for Y in MMSE sense.

"Proof:" Mean Squared Error

$\hat{y} = g(x)$

$\mathcal{E} = E[(y-\hat{y})^2]$

$= \iint (y-\hat{y})^2 P(x,y)\, dx\, dy$

$= \int dx \int dy\, (y-\hat{y})^2 P(x) P(y|x)$

$= \int dx\, P(x) \int dy\, (y-\hat{y})^2 P(y|x)$ ← $f(x)$.

$E[h(x,y)] = \iint h(x,y) P(x,y)\, dx\, dy$ (joint prob)

$P(x,y) = P(x)P(y|x) = P(y)P(x|y)$

$\Rightarrow$ To min $\mathcal{E}$ $\Rightarrow$ min $\int (y-\hat{y})^2$ Mean squared error $P(y|x)\, dy$ for each x.

$\sum a_i$, $a_i \geq 0$ min each a_i.

for each x, $\min \int dy\, (y-\hat{y})^2 P(y|x)$

$\frac{\partial}{\partial \hat{y}} \int dy\, (y-\hat{y})^2 P(y|x) = 0$

$-\int dy\, 2(y-\hat{y}) P(y|x) = 0$

$-\int dy\, (y-\hat{y}) P(y|x) = 0$

$\int \hat{y} P(y|x)\, dy = \int y P(y|x)\, dy$

$\hat{y} \int P(y|x)\, dy = \int y \cdot P(y|x)\, dy$

$\hat{y} = \int y \cdot P(y|x)\, dy = E[y|x]$

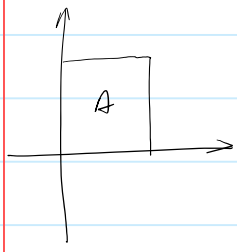
2

2017年10月16日 16:37

$E[(y - E[y|x])\, g(x)] = 0$ orthogonality Principle.

Example 1: $P_{xy}(x,y) = \begin{cases} c(x+y), & (x,y) \in A \\ 0, & (x,y) \notin A \end{cases}$

$c > 0$.



find the optimal estimate of y from x.

$= E[y|x]$

Example 2.

In practice:

$D \longrightarrow \{(x_1,y_1), (x_2,y_2) \cdots (x_n,y_n)\}$

$D \longrightarrow f(x) \approx E[y|x]$

① $D \xrightarrow[\text{density estimation}]{} \hat{P}(x,y) \longrightarrow E[y|x]$ | difficult when $\dim(x)$, $\dim(y)$ are high.

→ assume $P(x,y)$ is Gaussian | Problem: Data is not Gaussian

---

② find/start with a family of functions $y = g(x) = g(x|\theta)$ ← parameters.

$= \hat{y}(x|\theta)$

$\hat{\theta} = \arg\min \Sigma (y_i - g(x_i|\theta))^2/n$

$E[(y-\hat{y})^2]$

If $g(x|\theta)$ is rich

$\Rightarrow g(x|\hat{\theta}) \approx E[y|x]$

$\frac{\Sigma y_i}{n}$ = Sample mean $\xrightarrow[n\to\infty]{} E[Y]$

3

2017年10月23日 16:37

4. Linear and logistic regression

1) linear regression

function estimation:

$$D = \{(x_1, y_1), \cdots (x_n, y_n)\}$$

find: $f(\cdot)$ = linear.

$$\hat{y} = w^T x + b$$

$\hat{y}$: scalar; w^T: row vector; x: column vector; b: scalar

Example: x: 2dim, y: 1dim

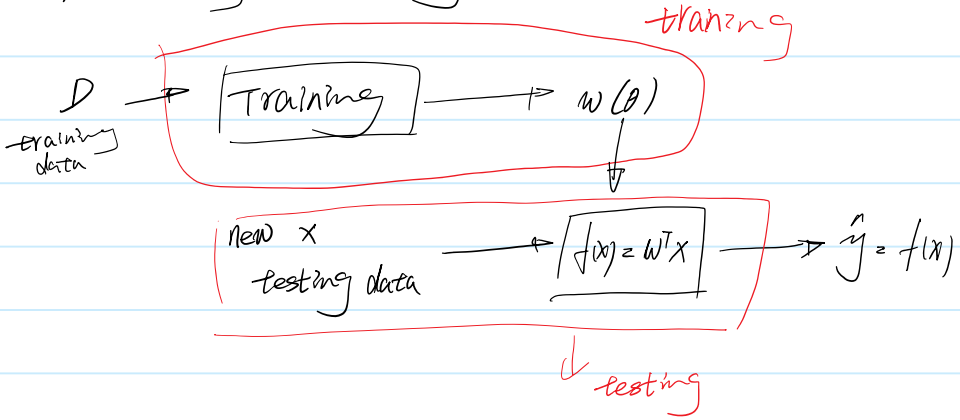
$$f(x) = w_1 x_1 + w_2 x_2 + w_0$$

$$= [w_1, w_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + w_0$$

$$= [w_1, w_2, w_0] \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}$$

$$\theta = [w_1, w_2, w_0]^T$$

2) Training (and testing)



"Training" $D \rightarrow w$

4

2017年10月23日 16:37

"Training" $D \longrightarrow w$

① 1st perspective / derivation, data-driven

recall $\mathcal{E} = MSE = \frac{\sum (y_i - f(x_i))^2}{n}$ (least squared) $\quad f(x) = w^T x$ ← augumented $\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ 1 \end{bmatrix}$

$= \mathcal{E}(w)$

optimal

$w^* = \arg\min_w \mathcal{E}(w)$

$\nabla_w \mathcal{E} = 0 \Rightarrow w^*$

$F(x) = F(x_1, x_2 \cdots x_n)$

$\nabla_x F = \frac{dF}{dx_1}$; $\frac{dF}{dx_n}$ = gradient

$\frac{d\mathcal{E}}{dw} = 0$ ⇓ normal equation

$\boxed{\nabla_w \Sigma = 0}$

Example: x: 1d

y: 1d.

$y = f(x) = w_1 x + w_0 = [w_1 \; w_0] \begin{bmatrix} x \\ 1 \end{bmatrix}$

$\mathcal{E} = \frac{1}{n} \sum \left(y_i - (w_1 x_i + w_0) \right)^2$

$\nabla_w \mathcal{E} = 0$, $\frac{\partial \mathcal{E}}{\partial w_1} = 0$ $\frac{\partial \mathcal{E}}{\partial w_0} = 0$

$\frac{\partial \mathcal{E}}{\partial w_1} = \frac{1}{n} \sum 2 \cdot \left(y_i - (w_1 x_i + w_0) \right) \cdot (-x_i) = 0 \Rightarrow \left(\sum x_i^2 \right) w_1 + \left(\sum x_i \right) w_0 = \sum x_i y_i$ (≈ $nE[x^2]$, ≈ $nE[x]$, ≈ $nE[xy]$)

$\frac{\partial \mathcal{E}}{\partial w_0} = \frac{-2}{n} \sum \left(y_i - (w_1 x_i + w_0) \right) \cdot 1 = 0 \Rightarrow \left(\sum x_i \right) \cdot w_1 + \left(\sum 1 \right) \cdot w_0 = \sum y_i$

5

2017年10月23日 16:37

$$\begin{bmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & \sum 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} \sum x_i y_i \\ \sum y_i \end{bmatrix}$$

$$A \cdot w = b$$

$$w^* = A^{-1} \cdot b$$

Another way.

$f(x) = w_1 x + w_0$

$$\varepsilon = E\left[(y - f(x))^2\right]$$

$$\frac{\partial \varepsilon}{\partial w_1} = 0 \qquad -E\left[2(y - (w_1 x + w_0)) \cdot x\right] = 0$$

$$E[x^2] w_1 + E[x] \cdot w_0 = E[xy]$$

$$\frac{\partial \varepsilon}{\partial w_0} = 0 \qquad -E\left[2(y - (w_1 x + w_0) \cdot 1\right] = 0$$

$$E[x] w_1 + 1 \cdot w_0 = E[y]$$

$$\begin{bmatrix} E[x^2] & E[x] \\ E[x] & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_0 \end{bmatrix} = \begin{bmatrix} E[xy] \\ E[y] \end{bmatrix}$$

$$\begin{bmatrix} \frac{\sum x_i^2}{n} & \frac{\sum x_i}{n} \\ \frac{\sum x_i}{n} & \frac{\sum 1}{n} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} \frac{\sum x_i y_i}{n} \\ \frac{\sum y_i}{n} \end{bmatrix}$$

6

2017年10月23日 20:18

General case x_i m-dim

y_i 1-dim.

$$E = \frac{1}{n} \sum (y_i - x_i^T w)^2$$ ← m+1 dim.

$$X = \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{pmatrix} \quad n \times m. \qquad y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

$$\nabla_w E = 0$$

$$\hookrightarrow (X^T X) w = X^T y$$

$$w = (X^T X)^{-1} X^T y$$

$m \times n$ $n \times m$ $m \times n$ $n \times 1$

$m \times m$

$m \times n \cdot n \times 1 \Rightarrow m \times 1$

Week9

2017年10月30日 20:18

ReView : Linear regression

$D = \{(x, y)\}$ $\quad D \rightarrow w$ (training)

$D \rightarrow f()$

$f(\cdot)$: $x \rightarrow y$

① least squares (data driven)
② random variable based.
③ Guassian based.

$\hat{y} = f(x) = w^T x$ (augmented)

Example : x_i : 1d samples $\rightarrow x$

y_i : 1d samples $\rightarrow y$

augmented. $w = \begin{bmatrix} w_1 \\ w_0 \end{bmatrix}$

find w : ① $\min \sum_{i=1}^{n} (y_i - w^T x_i)^2$

$\hat{y} = w_1 x_1 + w_0 = \begin{bmatrix} w_1 \\ w_0 \end{bmatrix} [x_1, 1]$

$\begin{bmatrix} \sum x_1^2 & \sum x_i \\ \sum x_1 & \sum 1 \ (n) \end{bmatrix} \begin{bmatrix} w_1 \\ w_0 \end{bmatrix} = \begin{bmatrix} \sum x_i y_i \\ \sum y_i \end{bmatrix}$

$E[X] = \frac{\sum [x_i]}{n}$

Assumes: $x_i \sim X$, $y_i \sim y$ ② $\min E[(y - w^T x)^2] \Rightarrow$ Normal Equ

$\begin{bmatrix} E[X^2] & E[X] \\ E[X] & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_0 \end{bmatrix} = \begin{bmatrix} E[XY] \\ E[y] \end{bmatrix}$

Assumes: X, y are Guassians ③ $\begin{bmatrix} x \\ y \end{bmatrix} \sim N\left(\begin{bmatrix} m_x \\ m_y \end{bmatrix}, \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{bmatrix}\right) \Rightarrow m_{y|x} = E[y|x]$

$= m_y + \sigma_{yx}(\sigma_x^2)^{-1}(x - m_x)$

$\hat{y} = f(x)$

$= \underbrace{\sigma_{yx}(\sigma_x^2)^{-1}}_{w_1} x + \underbrace{m_y - \sigma_{yx}(\sigma_x^2)^{-1} \cdot m_x}_{w_0}$

$m_x = E(x)$, $m_y = E[y]$

$\sigma_x^2 = E[(x - E[x])^2]$

$\sigma_y^2 = E[(y - E[y])^2]$

$\sigma_{xy} = \sigma_{yx} = E[(y - E[y]) \cdot (x - E[x])]$

Variance of Estimation error.

$\sigma_{y|x}^2 = \sigma_y^2 - \sigma_{yx}(\sigma_x^2)^{-1}\sigma_{xy}$

$\Downarrow$

$E[(y - E(y|x))^2]$

2

2017年10月30日 20:18

General Case x: m-dim Samples from x.
y: 1-dim. Samples from y

MSE $= \mathcal{E} = E[(y - w^T x)^2]$ (x augmented.)

$X = \begin{bmatrix} x_1 \\ \vdots \\ x_m \\ 1 \end{bmatrix}$ $W = \begin{bmatrix} w_1 \\ \vdots \\ w_m \\ w_0 \end{bmatrix}$

$\mathcal{E} = \frac{1}{n} \sum (y_i - x_i^T w)^2$

$D_w \mathcal{E} = 0$

$D_w \mathcal{E} = E[-2(y - w^T x) D_w(w^T x)]$

$= -2 E[(y - w^T x) \cdot x] = 0$

$E[(w^T x) \cdot x] = E[yx]$

$= E[xy]$

($(w^T x)$: scalar; xy: column vector $\cdot$ scalar)

$\Rightarrow E[(x^T w) \cdot x] = E[xy]$

$E[x \cdot (x^T w)] = E[xy]$

($(m+1) \times 1$ $\quad$ $1 \times (m+1)$)

$w E[x x^T] = E[xy]$

$w = (E[x x^T])^{-1} E[xy]$

Approximation (using D)

$E[xy] = \frac{1}{n} \sum x_i y_i$

$E[x x^T] = \frac{1}{n} \sum x_i x_i^T$ (x augmented.)

$[\,] * [\,] \Rightarrow [\,]$

Gradient (variable vector)

$D_x(a^T x)$ (constant vector)

$= D_x(\sum a_i x_i) = \begin{bmatrix} \frac{\partial}{\partial x_1} \sum a_i x_i \\ \frac{\partial}{\partial x_2} \sum a_i x_i \\ \vdots \\ \frac{\partial}{\partial x_n} \sum a_i x_i \end{bmatrix}$

$= \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a \end{bmatrix} = a.$

$D_x(a^T x) = a$

$D_x(x^T a) = a.$

3

2017年10月30日 20:18

$$X = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{bmatrix} = \begin{bmatrix} [\; \cdots x_1 \cdots \; 1] \\ [\; \cdots x_2 \cdots \; 1] \\ \vdots \\ [\; \cdots x_n \cdots \; 1] \end{bmatrix} \quad (m+1)$$

$$X^T = \left[\begin{bmatrix} \\ x_1 \\ \\ 1 \end{bmatrix} \begin{bmatrix} \\ x_2 \\ \\ 1 \end{bmatrix} \cdots \begin{bmatrix} \\ x_n \\ \\ 1 \end{bmatrix} \right]$$

$$= [x_1, x_2 \cdots x_n]$$

$$\frac{1}{n} \sum x_i x_i^T = X^T X$$

$$X, X^T \Rightarrow = [x_1, x_2 \cdots x_n] \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n \end{bmatrix} = \sum x_i x_i^T$$

$$\sum x_i y_i = [x_1, x_2, \cdots x_n] \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = X^T Y$$

$$\frac{1}{n}(X^T X) w = \frac{1}{n} X^T Y$$

$$w = (X^T X)^{-1} (X^T Y)$$

---

3) Regularized linear regression

① why ?

Data → training → $w = (X^T X)^{-1} X^T y$ → training error

Data → testing → $\hat{y} = w^T x$ → testing error

Possible reasons:

$(X^T X)$ ill-conditioned

Possible solution: "Regularization"

② a common regularization: l_2 minimization.

$$w = \arg\min_w E[(y - w^T x)^2]$$

$$\Rightarrow w = \arg\min_w \{ E[(y - w^T x)^2] + \lambda w^T w \}$$

$$\|w\|_2^2 = \sum_{j=0}^{m+1} w_j^2$$

$\Downarrow$

$f(w)$

4

2017年10月30日 20:18

Min; Gradient

$$f(w) = \mathcal{E}(w) + \lambda w^T w \qquad \lambda > 0$$

$$D_w F(w) = D_w \mathcal{E}(w) + \lambda D_w (w^T \cdot w) \qquad (A = I)$$

$$= E[(y - w^T x) \cdot x] + \lambda \cdot 2I \cdot w$$

$$\Rightarrow \quad (E[xx^T] + 2\lambda I) w = E[xy]$$

makes more invertable.

$D_x \; x^T A x$ ($1 \times m$; $m \times m$ square & symmetrical; $m \times 1$)

$= 2Ax$

2.2) variation: L_1 minimization $\quad w^T w \rightarrow \|w\|_1 = \sum_{j=0}^{m} |w_j|$

2.3) PCA analysis

$X_{m \times 1} \rightarrow X_1 \rightarrow LR.$

$\rightarrow [\;]$

Week10

2017年11月6日 20:18

4) Iterative Solution : Gradient descent

① why?

LR equation : $W^* = (E[XX^T])^{-1} E[Xy]$

$m \times m$

$\dim(X) = m$.

if m is very large

$\Rightarrow (E[XX^T])^{-1}$ computationally expensive

Solution

① PCA dim reduction.

$X \rightarrow [PCA] \rightarrow X'$

m dim $\quad m'$ dim.

only work if most of EV of $E[XX^T]$ is small

e.g. $\sim 10^4 \quad \sim 10^1$

② Iterative Solution

$W_{n+1} = W_n + \Delta W_n$.

W_n: result from previous iteration

ΔW_n: increment

Gradient Descent

Newton Method.

Pseudo + limited memory

2017年11月6日 20:18

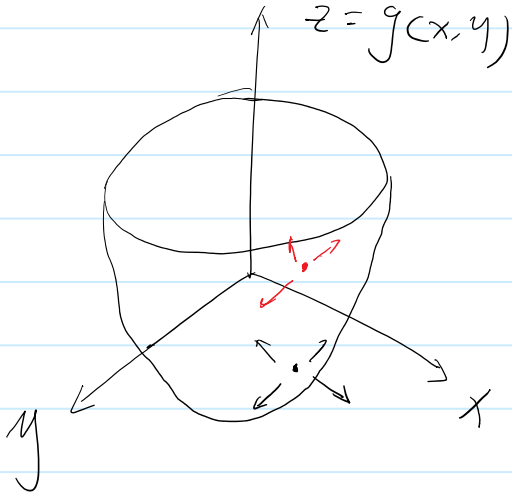
② Gradient Descent

Gradient Descent:

$$\Delta w_n = -\varepsilon \nabla_w E(w)\Big|_{w=w_n}$$

(minimized; Gradient)

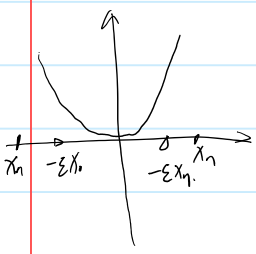
Recall: $E(w) = E\left[(y - w^T x)^2\right]$



Example GD

$g(x) = x^2$

$$\nabla g = \frac{dg}{dx} = 2x$$



$$x_{n+1} = x_n - \varepsilon 2x_n = x_n - \varepsilon x_n.$$

$$x_{n+1} = x_n - \varepsilon x_n = (1-\varepsilon)x_n$$
$$= (1-\varepsilon)(1-\varepsilon)x_{n-1}$$
$$= (1-\varepsilon)^n x_0$$

If $0 < \varepsilon < 1 \rightarrow x_n \rightarrow 0$ as $n \rightarrow \infty$

$$\frac{dg}{d\vec{n}} = \vec{n} \cdot \nabla g \quad \text{(gradient)}$$

$\max\left|\frac{dg}{d\vec{n}}\right|$ achieved when $\vec{n}$ in the same/opp. direction

descent → opp direction of ∇g.

GD: $x_{n+1} = x_n - \varepsilon \nabla g(x_n)$

3

2017年11月6日 20:18

GD: $x_{n+1} = x_n - \varepsilon \nabla g(x_n)$

applied to linear regression

$$\mathcal{E}(w) = E\left[(y - w^T x)^2\right]$$

$$\nabla_w \mathcal{E}(w) = -2\, E\left[(y - w^T x)\cdot x\right]$$

$$= 2\left(E[xy] - E[xx^T]w\right)$$

GD: $w_{k+1} = w_k - \varepsilon\left(E[xy] - E[xx^T]\, w_k\right)$ (ε: learning rate)

in each iteration

Take a random subset of $\{x_i, y_i\}$ from entire training data.

$E[xy] = \frac{1}{n_1}[x_i y_i]$ over subset

$E[xx^T] = \frac{1}{n_1}\sum x_i x_i^T$

$\Rightarrow$ min batch.

Practical implementation:

In each iteration

$$E[xy] = \frac{1}{n}\sum_{i=1}^{n} x_i y_i, \quad E[xx^T] = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^T$$

$\Rightarrow$ Batch method

n: sample size

In each iteration:

Take one pair x_i, y_i, $E[xy] = x_i y_i$, $E[xx^T] = x_i x_i^T$

$\Rightarrow$ SGD / LMS / adaptive filtering

Stochastic Gradient Descent

4

2017年11月6日 20:18

5) logistic regression

① Problem: to learn a linear classifier

Data: $D = \{(x_i, y_i)\}$

x_i: n-dim

y_i: 1 dim $y_i \in \{0, 1\}$

H_0, H_1

$D \rightarrow g(x) = P[y=1]$

$D \rightarrow g(\cdot)$ $\quad g(x) = [y(x)=1]$ $\Rightarrow$ $D \rightarrow w$

$\Rightarrow$ learning:

$g(x) = \frac{1}{1+e^{-w^T x}}$

$= \varphi(w^T x)$

$E(w)$: function to be optimised

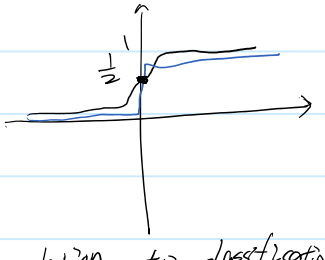
cross entropy (no-linear non quadradiet)

GD: $w_{n+1} = w_n - \varepsilon \nabla_w E(w_n)$ (loss function)

---

② logistic function: Continous version of threshold.

$\varphi(t) = \frac{1}{1+e^{-t}}$

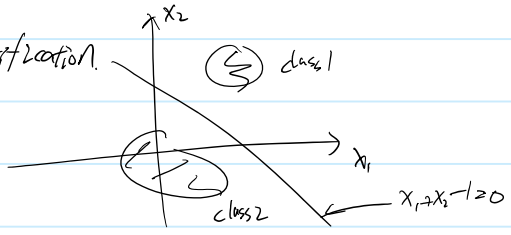


$\mu(t) = \begin{cases} 0, & t<0 \\ 1, & t \geq 0 \end{cases}$

$\Downarrow$

dosent have derivative.

relation to classification



$\varphi(w^T x)$ generates $P[X \in H_1]$

Soft classification

Surpassed by

- linear: SVM (kernal)
- non-linear: Nueral Network

5

2017年11月6日 20:18

④ Cross entropy 交叉熵

X: $P(x)$ $\quad -\log_2 P(x) = \overset{\min}{\#}$ of bits needed to encode x.

X	1	2	3	4
$P(x)$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$

1 → 2 bits 00
2 → 2 bits 01 ave # of bit need = 2
3 → 2 bits 10
4 → 2 bits 11

X	1	2	3	4
$P(x)$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{8}$
bits	1	2	3	3
	0	10	110	111

ave: $1\times\frac{1}{2} + 2\times\frac{1}{4} + 3\times\frac{1}{8} + 3\times\frac{1}{8} = \frac{1}{8}(4+4+3+3)$
$= \frac{7}{4} < 2$

$\sum_x P(x)\left(-\log P(x)\right) = H(x)$ = min ave # bits need to code x.
= the amount of uncertainty
= entropy

Cross: density { real ∝ assumed.

6

2017年11月6日 20:18

7

2017年11月6日 20:18

Week11

2017年11月13日 20:18

Cross entropy:

Two possibilities X: binary

$P = P[X=1]$ $\quad 1-P = P[X=0]$

$q = Q[X=1]$ $\quad 1-q = Q[X=0]$

$$H(P,q) = -P\log q - (1-P)\log(1-q)$$

$$H(P) = -P\log P - (1-P)\log(1-P)$$

In general

X	x_1	x_2	$\cdots$	x_k
P	p_1	p_2	$\cdots$	p_k

X	x_1	x_2	$\cdots$	x_k
q	q_1	q_2	$\cdots$	q_k

$$H(P) = -\sum p_k \log p_k$$

= min. ave. codeword length need

$$H(P,q) = -\sum_{k=1}^{k} p_k \log q_k$$

= ave. codeword length by assuming the data distribution to be q while it actually is P

(optimize codeword length according to q)

Example:

$P = [\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}]$ $\quad H(P) = \frac{1}{2} + 2\cdot\frac{1}{4} + 3\cdot\frac{1}{8} + 3\cdot\frac{1}{8} = \frac{7}{4} = 1.75$

$q = [\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}]$ $\quad H(q) = 2$

$$H(P,q) = -\sum P_k \log q_k = -\left[\frac{1}{2}(-2) + \frac{1}{4}(-2) + \frac{1}{8}(-2) + \frac{1}{8}(-2)\right]$$

$$= 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{4} = 2 > H(P) = 1.75$$

$$H(P,q) - H(P) = 2 - 1.75 = 0.25$$

$q = [\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}]$

$P = [\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}]$

$H(P,q) = \frac{7}{4} > H(P) = 2.$

$H(P,q) - H(P) \geq 2$

$D(P\|q)$: KL distance

$$D(P\|q) = \sum P_k \log \frac{P_k}{q_k} \neq D(q\|P)$$

2017年11月13日 20:18

Application in estimation

p: unknown distribution

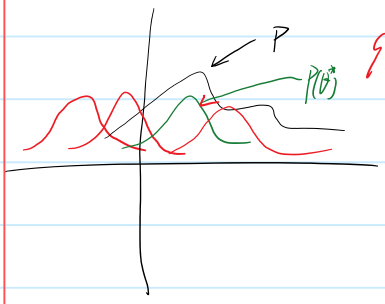
q: parametric distribution

$q = q(\theta)$

find an estimate of p using $q(\theta)$.

$$\theta^* = \arg\min_\theta D(p \| q(\theta))$$

$q(\theta^*)$ best estimate of p.



$$E[X] = \int_{-\infty}^{+\infty} x\, p(x)\, dx \approx \frac{1}{n}\sum x_i$$

$$\theta^* = \arg\min_\theta D(p \| q(\theta)) = \arg\min_\theta H(p, q(\theta))$$

$$D(p\|q) = H(p,q) - H(p)$$

find a q that most close to p

---

④ Apply cross entropy to logistic regression

recall linear regression:

$$y = h(x) = w^T x$$

$$MSE = E[(y - w^T x)^2] \rightarrow w = E[xx^T]^{-1} E[xy]$$

logistic regression:

two class classification

x — input, m-dim

y — output, 1-dim $y \in \{0, 1\}$

$$P(x) = P[y=1|x]$$

= true distribution = unknown.

$q(x)$ = approximation to $p(x)$

$$= \varphi(w^T x) = \frac{1}{1 + e^{-w^T x}}$$

↖ augmented

$H(p,q)(x)$

$= [-p \log q - (1-p)\log(1-q)](x)$

logistic regression: find $q(x)$: best appro. to $p(x)$.

$$\Rightarrow w^* = \arg\min_w H(p,q)(x) = \arg\min_w E[H(p,q)(x)]$$

3

2017年11月13日 20:18

$H(p,q)(x)$

$= [-p \log q - (1-p)\log(1-q)](x)$

logistic regression: find $q(x)$ = best approx. to $p(x)$.

$\Rightarrow w^* = \arg\min_w H(p,q)(x) = \arg\min_w E[H(p,q)(x)]$

$q(x) = \frac{1}{1+e^{-w^T x}}$

$\Rightarrow$ GD $w_{n+1} = w_n - \varepsilon \nabla_w E[H(p,q)(x)]$

$\nabla_w E[H(p,q)(x)]$

$= E[(q-p)x]$

$\approx \frac{1}{n}\sum (q(x_i) - p(x_i))x_i$

$\approx \frac{1}{n}\sum (q(x_i) - y_i)x_i$

$p(x) = P[y=1|x]$

$y_i = 1 \rightarrow p(x_i) \simeq 1$

$y_i = 0 \rightarrow p(x_i) = 0$

4

2017年11月13日 20:18

⑤ multi class case

input X : m-dim

output y : 1-dim $y \in \{1, 2, \cdots k\}$

k-dim indicator vector $y = [y_1, y_2 \cdots y_k]$

$y = k^{th}$ class $= [0, 0 \cdots \underset{k}{1}, 0 \cdots 0]$

$P(X) = [P_1(X), P_2(X) \cdots P_k(X)]^T$

$P_k(X) = P[y = k^{th} \text{ class} \mid x]$

$q(X) = [q_1(X), q_2(X) \cdots q_k(X)]^T$ based on $q(y)$

$$q_k(X) = \frac{e^{+w_k^T X}}{\sum_{k'} e^{+w_{k'}^T X}}$$ ← $q(k)$; denominator ← normalization

$\hat{y}$ = class k if $k = \arg\max_k q_k(X)$ Soft max regression

$$\frac{e^{-w_k^T X}}{\sum_{k'=1}^{k} e^{-w_{k'}^T X}}$$

$$H(P, q)(X) = \left(-\sum_{k=1}^{k} P_k \log q_k\right)(X)$$

$$w^* = \arg\min_w E\left(H(P, q)(X)\right)$$

$(w_1^* \; w_2^* \cdots w_k^*)$

↑ $m+1$-dim.

$(\sum_{k=1}^{k} q_k = 1)$

$$w_{k, n+1} = w_{k, n} - \varepsilon \nabla_{w_k} E[H(P, q)(X)]$$

$$\nabla_{w_k} E[H(P, q)(X)] = E[(q_k - P_k) X]$$

$$\approx \frac{1}{n} \sum \left(q_k(X_i) - P_k(X_i)\right) X_i$$

$$\approx \frac{1}{n} \sum (q_k(X_i) - y_i) X_i$$

:

Ex $q(X) = [0.1, 0.2, 0.3, 0.15, 0.25]$

$\Rightarrow \hat{y} = 3$, 3rd class

:

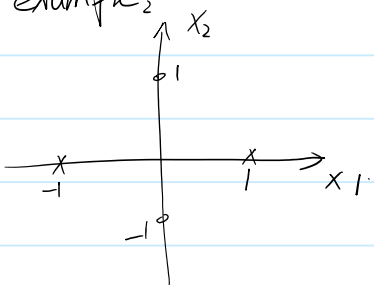
2017年11月13日 20:18

5. Random forest and Boosting

1) CART (classification and regression trees)

① classification tree: use a tree to represent a classifier

example:

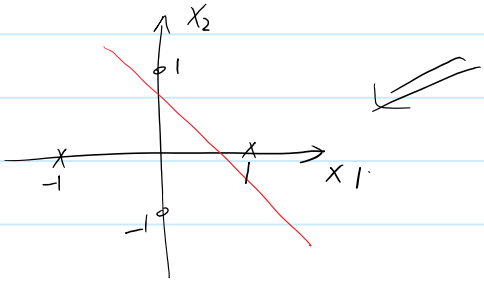


classifier: $y = f(x)$.

input x, m-dim $m = 2$.

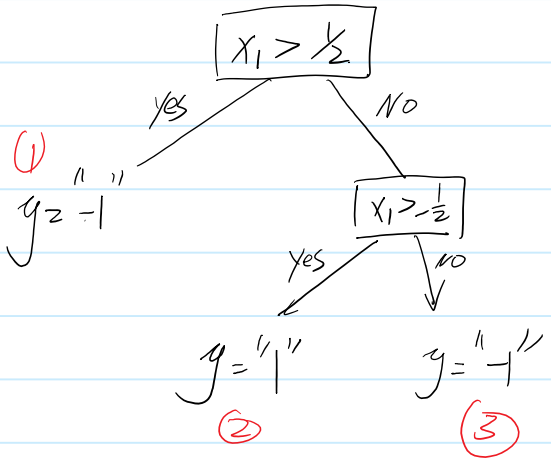
output y, 1-dim $y \in \{-1, +1\}$

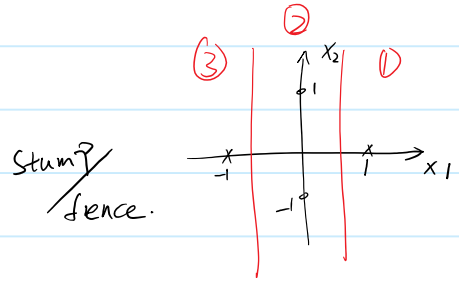
X O



past: $y = \text{sign}(w^T x)$ | now: Tree / Decision tree / classifier

$f(\cdot)$:





Stump? / fence.

$$f(x) = \sum_i a_i I_{A_i}(x)$$ ← set indicator function.

$$I_A(x) = \begin{cases} 0, & x \in A \\ 1, & x \notin A \end{cases}$$

$$= (-1) I_{A_1}(x) + (+1) I_{A_2}(x) + (-1) I_{A_3}(x)$$

6

2017年11月13日 20:18

$f(x) = 2x - 4$

$x_{n+1} = x_n - \epsilon(2x_n - 4)$

$= (1-2\epsilon)x_n + 4\epsilon$

$= (1-2\epsilon)^2 x_{n-1} + 4\epsilon + 4\epsilon(1-2\epsilon)$

$= (1-2\epsilon)^n x_0 + 4\epsilon(1+(1-2\epsilon)\cdots(1-2\epsilon)^{n-1})$

$= (1-2\epsilon)^n x_0 + 4\epsilon \cdot \frac{1-(1-2\epsilon)^{n+1}}{2\epsilon})$

$\because -1 < 1-2\epsilon < 1$

when $n \to +\infty$ $(1-2\epsilon)^n$ or $(1-2\epsilon)^{n-1} \to 0$

$\therefore x_{n+1} \simeq 0 + 4\epsilon \cdot \frac{1}{2\epsilon}$

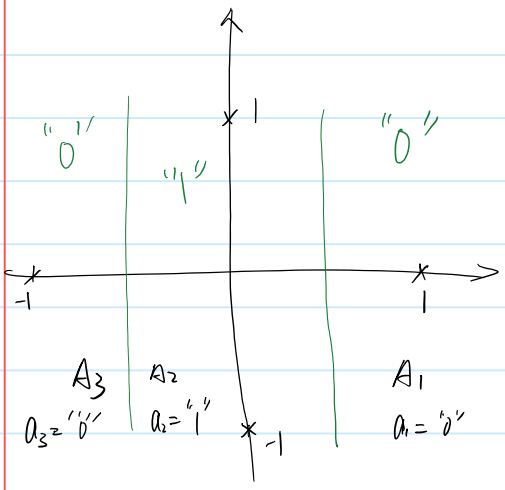
$= 2$

Week12

2017年11月20日 20:18

Review: CART (classification & Regression tree)

(highly nonlinear classification in CS way)



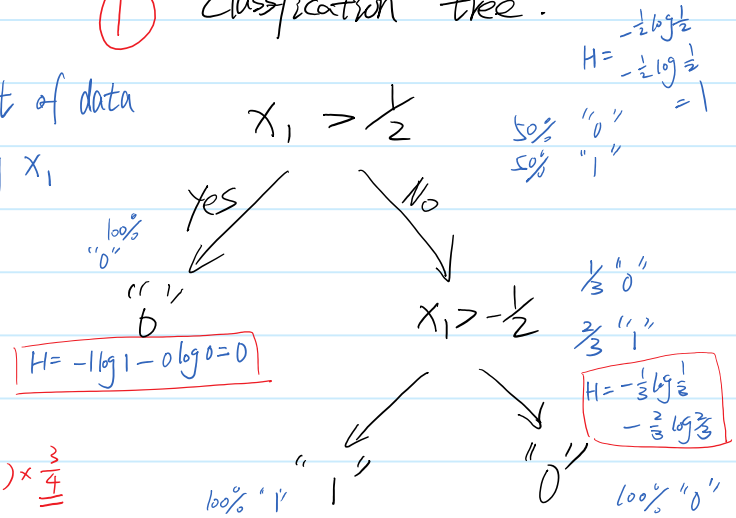
tree ⇕ partition of input space

each subset has a unique label

① classification tree.

(1) split of data

(2) along x_1

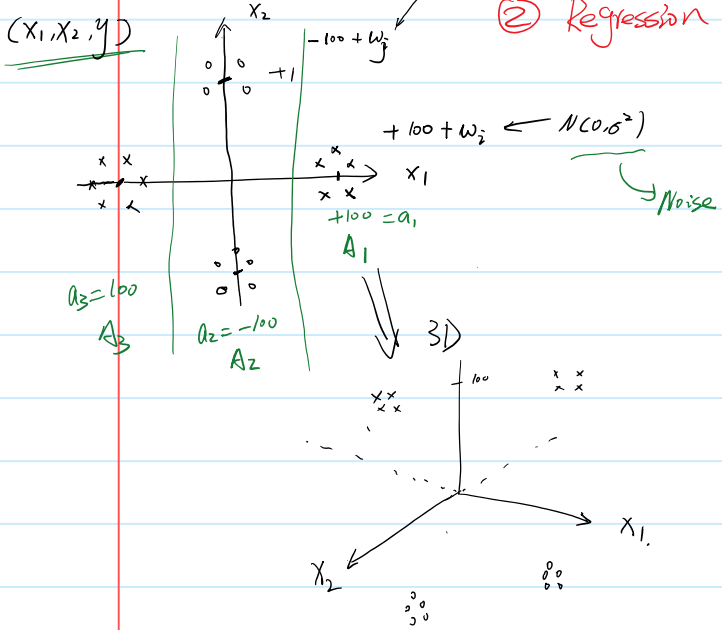


ave: $0\cdot\frac{1}{4} + (\ )\times\frac{3}{4} \leq 1$ → previous layer H

IDEA: reduce entropy

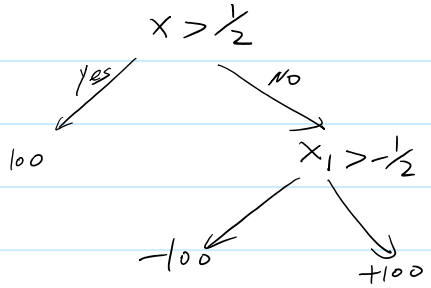
$f(x) = \sum a_i I_{A_i}(x)$

② Regression Tree.



regression function

$\hat{y} = f(x)$



$f(x) = \sum a_i I_{A_i}(x)$

2017年11月20日 20:18

③ Training for CART.

$$D = \{(x_i, y_i)\} \Longrightarrow f(\cdot)$$

dataset D $\quad F(D)$

(1) classification, $F(D)$: "purity"

$\hat{P}_k$ = % of point (x_1, x_2) that belong to class k ($k = 0, 1$)

$F(D) = F(\{\hat{P}_k\})$

$= H(\{\hat{P}_k\})$

$= -\hat{P}_k \log \hat{P}_k - (1-\hat{P}_k) \log (1-\hat{P}_k)$

(2) Regression

$D = \{(x_1, y_1) \cdots (x_m, y_m)\}$

$f(x) = \frac{1}{m} \sum y_i$

$= \arg\min_a F(D)$

$F(D) = \min_a \sum_m (y_i - a)^2$.

⇒ Choose "a" to represent y_i.

a value

3

2017年11月20日 17:53

2) Random forest.

① Ensemble learning

independent

$f_i(\,)$: classifier i. $P_e < 50\%$

majority vote classifier = majority $\{\{f_i()\}\}_{i=1}^{2n+1}$ ← odd number.

AS $n \to \infty$, Prob error $\to 0$

P_e (majority vote)

Binomial distribution, $= \sum_{k=n+1}^{2n+1} C_{2n+1}^{k} \varepsilon^k (1-\varepsilon)^{2n+1-k}$ k out of 2n+1 are in error.

$\varepsilon = P_e$ individual classifier. ex: 49%

② Apply to CART

$D = \{(x_i, y_i)\}$ → random trees. $\{T_m\}$

How to generate random trees

classification

$T(X) = \text{Majority}\,(T_m(X))$

Regression

$T(X) = \frac{1}{M} \sum_{m=1}^{M} (T_m(X))$

4

2017年11月20日 17:53

Random Trees: increase the randomness between individual classifier variables

Tm( ):
- Take a random subset of D → along x_1 or x_2
- when making a split decision
  Only look at p_1 random variables out of total p variables

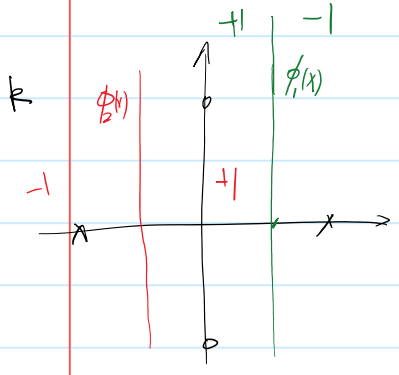
$x_1\ x_2\ x_3\ \cdots\ x_p$ → p_1 variable. → split

3) Boosting

① Basic Idea: $P_c > 50\%$

- a sequence of weak learners:
- Combined (adding) in a strong learner
  $P_c = 100\%$ (on training data)

Example: classification



$\phi_1(x)$ (classifier 1)
$= \begin{cases} -1, & x \geq \frac{1}{2} \\ +1, & x < \frac{1}{2} \end{cases}$

$P_e = \frac{1}{4} = 25\%$

$P_c = \frac{3}{4} = 75\% > 50\%$

(try to previous mistake)

$\phi_2(x)$ (classifier 2).
$= \begin{cases} -1, & x \leq -\frac{1}{2} \\ +1, & x > -\frac{1}{2} \end{cases}$

$P_e = \frac{1}{4} = 25\%$

$P_c = \frac{3}{4} = 75\%$

Boost

$f(x) = \phi_1(x) + \phi_2(x) = \begin{cases} 0, & x \geq \frac{1}{2} \\ 2, & -\frac{1}{2} \leq x \leq \frac{1}{2} \\ 0, & x \leq -\frac{1}{2} \end{cases}$

$P_c = 100\%$

Week14

2017年12月4日 17:53

Review boosting

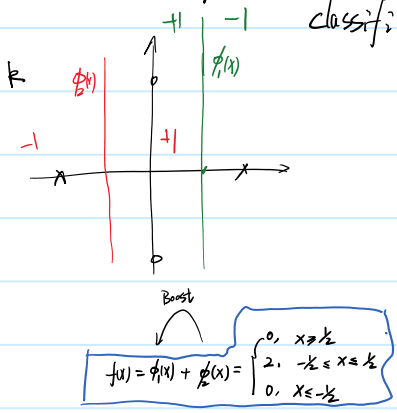
Basic idea.

$f(x)$ = classifier / regression function

$$f(x) = \sum \alpha_m \phi_m(x)$$

ϕ_m: weak learner ($P_e < 50\%$)

Example: classification



$\phi_1(x)$ (classifier 1)

$$= \begin{cases} -1, & x \geq \frac{1}{2} \\ +1, & x < \frac{1}{2} \end{cases}$$

$P_e = \frac{1}{4} = 25\%$

$P_C = \frac{3}{4} = 75\% > 50\%$

(Try to previous mistake)

$\phi_2(x)$ (classifier 2).

$$= \begin{cases} -1, & x \leq -\frac{1}{2} \\ +1, & x > -\frac{1}{2} \end{cases}$$

$P_e = \frac{1}{4} = 25\%$

$P_C = \frac{3}{4} = 75\%$

② Ada Boost.

adaptive

$f(x)$: classification function.

$f_m(x) \longrightarrow f(x)$

$$f_m(x) = f_{m-1}(x) + \alpha_m \phi_m(x, \gamma_m)$$

$$\phi_m(x, \gamma_m) = \begin{cases} -1, & x > \frac{1}{2} \\ +1, & x < \frac{1}{2} \end{cases}$$

$\gamma_m = [\frac{1}{2}, 0]^T$

$\alpha_m = 1$

2017年12月4日 17:53

How to find α_m, γ_m.

Given: $f(x)$, ϕ_m, C

$$(\alpha_m, \gamma_m) = \underset{\alpha,\gamma}{\operatorname{argmin}} \mathcal{L}(y, f_m(x))$$

$\mathcal{L}$ → loss function.

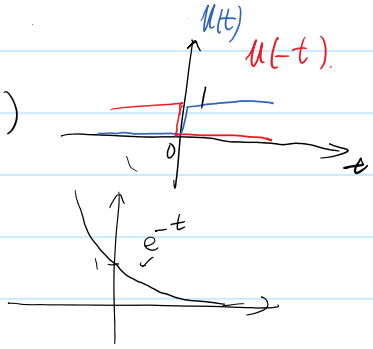
$y = f(x)$ = out of classifier (truth) $\in \{-1, +1\}$

$f_m(x)$ = "boost" classifier $= f_{m-1}(x) + \alpha \phi_m(x, \gamma)$ $\in \{-1, +1\}$

$\mathcal{L}(y, f_m(x))$ = loss function

ideally $= \begin{cases} 0, & y = f_m(x) \\ 1, & \text{o.w.} \end{cases} = u(-y f_m(x))$

not derivative



$\approx e^{-y \cdot f_m(x)}$

② Ada Boost Optimization.

$$E[\mathcal{L}(y, f_m(x))] = \mathcal{L}_m = \frac{1}{n} \cdot C \cdot \sum_{i=1} \mathcal{L}(y_i, f_m(x_i)) \qquad D = \{(x_i, y_i)\}$$

$$= C \cdot \sum e^{-y_i f_m(x_i)}$$

$$= C \cdot \sum e^{-y_i f_{m-1}(x_i) - y_i \alpha \phi(x_i, \gamma)}$$

α: alpha.
γ: Gamma.

see textbook

$\underset{\alpha,\gamma}{\operatorname{argmin}} \mathcal{L}_m \longrightarrow \alpha_m, \gamma_m$.

basic idea, optimize α first and optimize γ 2nd

2017年12月4日 17:53

④ Gradient Boosting → See in Elements of statistical learning

In classification / regression.

$D_{(\text{training data})} = \{(x_i, y_i)\}$

$\longrightarrow f(x)$

y_i: output / ground truth.

Want to find $f(x)$ s.t.

$E[L(y, f(x))]$ is minimized.

$E[L(y, f(x))] = \frac{1}{n} \sum L(y_i, f(x_i))$ ← function

$f^*(\,) = \underset{f(\,)}{\arg\min}\ E[L(y, f(x))]$

function variate × / approximate ✓

$\vec{f} = [f(x_1), f(x_2), \cdots f(x_n)]^T$

$\vec{y} = [y_1, y_2 \cdots y_n]^T$

$E[L(y, f(x))]$ — not known but know what it should be.

$\approx \frac{1}{n} \sum L(y_i, f(x_i))$

$= \frac{1}{n} \sum L(y_i, f_i)$

$= L(\vec{y}, \vec{f})$

find $\vec{f}$ by GD

$$\vec{f}_m = \vec{f}_{m-1} - \varepsilon_m \nabla_{\vec{f}} L \Big|_{\vec{f} = \vec{f}_{m-1}}$$

2017年12月4日 17:53

find $\vec{f}$ by GD

$$\vec{f}_m = \vec{f}_{m-1} - \varepsilon_m \nabla_{\vec{f}} \mathcal{L}\big|_{\vec{f}=\vec{f}_{m-1}}$$

$$\vec{f}_m \xrightarrow[m\to\infty]{} \vec{f}^* \qquad \vec{f}^* = [\vec{f}^*(x_1) \cdots \vec{f}^*(x_n)]^T$$

optimizing values of function not function.

new $x \longrightarrow f^*(x) = ?$

& $\{x_1, x_2 \cdots x_n\}$

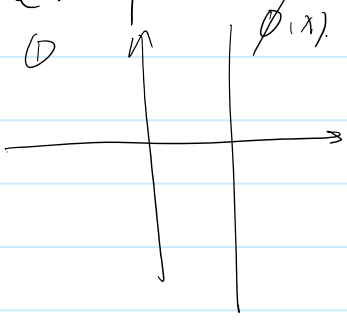
extra process ⟶ interpreting the gradient function

introduce function.

$$\phi(x) = \phi(x; \gamma)$$

↑ parameters

example.



ϕ_i 们 not need to be Gaussian can be any function.

② $\phi(x;\gamma) = C \cdot e^{-(x-m)^2/2\sigma^2}, [m, \sigma^2]^T = \gamma$

minimize gradient g & $\phi(x;\gamma)$ are points x_i

$$\gamma_m = \arg\min_{\gamma} \sum \mathcal{L}(g_i, \phi(x_i, \gamma))$$

$$g_i = \nabla_f \mathcal{L}(\vec{y}, \vec{f})(\vec{f}=\vec{f}_{m-1})$$

$$\Rightarrow f_m(x) = f_{m-1}(x) - \varepsilon \phi(x, \gamma_m)$$

example,

$$\mathcal{L}(y, f(x)) = (y - f(x))^2 \Rightarrow L_2 \text{ boost}$$

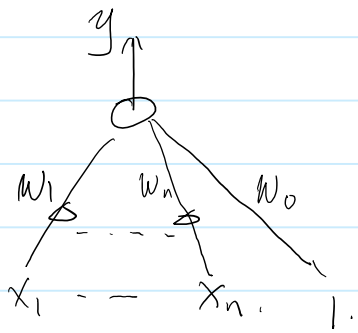
5

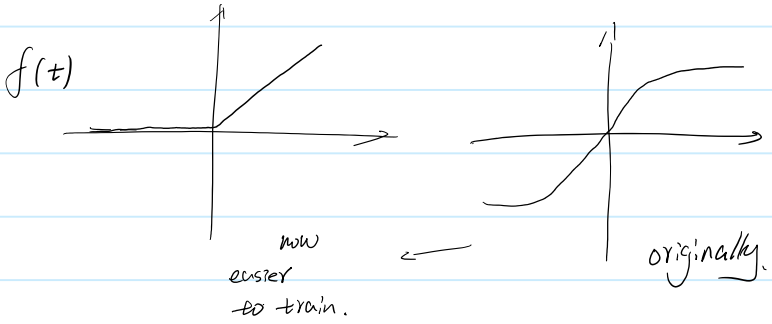
2017年12月4日 17:53

Neural Networks

1) a "neuron" $y = f(w_0 + w_1 x_1 + \cdots + w_n x_n)$.

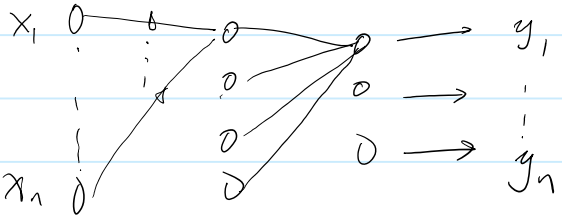
$= f(w^T x)$





2) A neural network

approximate a function with multiple inputs & outputs.



3) Training:

$D = \{(x_i, y_i)\}$ ⟶ a neural network (fixed structure)

$\Downarrow f(x)$

neural network $= f(x) = f(x_i; w)$ ← weights

training: $W_t = W_{t-1} - \varepsilon \nabla_w \sum L(y_i, f(x_i)) \Big|_{w = w_{t-1}}$

⇒ backpropagation

kingma

Adam algorithm.

6

2017年11月20日 17:53

Problem:

① converge is to slow

② converge to a plane ?

pulseo — newton technique // not memory efficient

Week2

2018年1月29日 18:00

Review

Machine Learning

- classification $x \xrightarrow{f(\cdot)}$ label / class
- Regression $x \xrightarrow{f(\cdot)} y$

$P[X \in \text{class } k] = f(X)$

Gaussian Distribution.

$X \sim N(m, \Sigma)$. $P(X) = (2\pi)^{-\frac{n}{2}} |\Sigma|^{-\frac{1}{2}} e^{-\frac{1}{2}\|X-m\|^2_{\Sigma}}$

$(x-m)^T \Sigma^{-1} (x-m)$

Marginal & Conditional

Marginal: $\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix} \begin{matrix} X_1 \\ \\ \\ X_2 \end{matrix}$ $X_1 \sim N(m_1, \Sigma_1)$

$\begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_n \end{bmatrix}$ $\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$

Conditional:

$X_1 | X_2 \sim N(m_{1/2}, \Sigma_{1/2})$

$m_{1/2} = m_1 + \Sigma_{12}\Sigma_{22}^{-1}(X_2 - m_2)$

$\Sigma_{1/2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$

$P(X_1|X_2) = \frac{P(X)}{P(X_2)}$ ← $P(X_1, X_2)$

$= \frac{(2\pi)^{-\frac{n}{2}} |\Sigma|^{-\frac{1}{2}} e^{-\frac{1}{2}\|X-m\|^2_{\Sigma}}}{(2\pi)^{-\frac{n}{2}} |\Sigma_{22}|^{-\frac{1}{2}} \cdot e^{-\frac{1}{2}\|X_2-m_2\|^2_{\Sigma_{22}}}}$ $\Rightarrow \|X_1 - m_{1/2}\|^2_{\Sigma_{1/2}}$ ← X_2

$f(u,v) = u^2 + auv + v^2 + bu + cv + d$. $\frac{\partial f}{\partial u} = 2u + av + b = 0$ $u = \frac{-(av+b)}{2}$ ⇓ $m(v)$

$f(u,v) = (u - m(v))^2 + C(v)$ $\frac{\partial f}{\partial u} = 2(u-m) = 0 \Rightarrow u = m$

2018年1月29日 18:00

4) Optimal classification of Gaussian Distribution

① Two class problem. (hard classifier)

$H_0: x \sim N(m_0, \Sigma_0)$

$H_1: x \sim N(m, \Sigma_1)$

optimal classifier

$$\log \frac{P(x|H_1)}{P(x|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \log \frac{P_0}{P_1}$$

L: LHR

log likelihood ratio

$$LHS = \log \frac{(2\pi)^{-n/2} |\Sigma_1|^{-1/2} e^{-\frac{1}{2}\|x-m_1\|^2_{\Sigma_1}}}{(2\pi)^{-n/2} |\Sigma_0|^{-1/2} e^{-\frac{1}{2}\|x-m_0\|^2_{\Sigma_0}}}$$

$$= C - \frac{1}{2}\left(\|x-m_1\|^2_{\Sigma_1} - \|x-m_0\|^2_{\Sigma_0}\right)$$

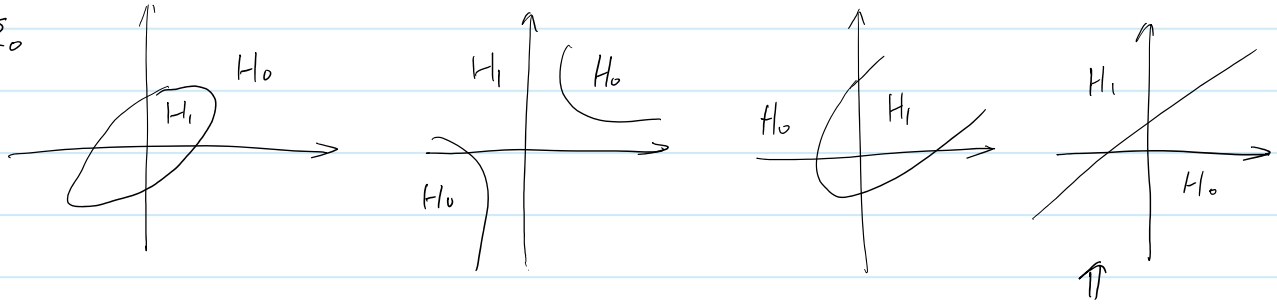
= quadratic in x

Boundary x, LHS = RHS

$$\|x-m_1\|^2_{\Sigma_1} - \|x-m_0\|^2_{\Sigma_0} + C' = 0$$

Quadratic in x.

Shape Depends on Σ_1, Σ_0

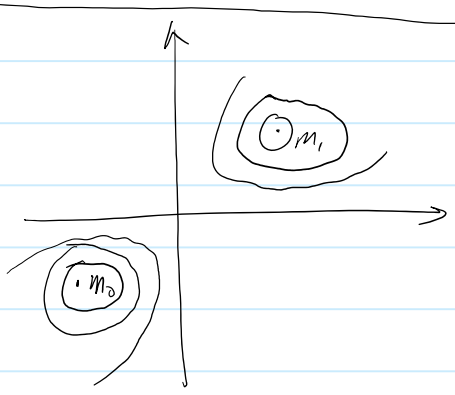


Special case

Special case.

① $P_0 = P_1 = 1/2$

$\Sigma_0 = \Sigma_1 = \sigma^2 I$.



$$\|x-m\|^2_{\Sigma} = \frac{1}{\sigma^2}\|x-m\|^2.$$

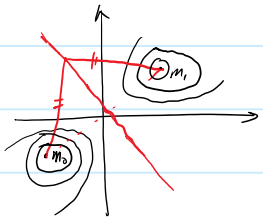
3

2018年1月29日 18:00

Note: "−" sign in front.

$$\|x-m_1\|^2 \underset{H_1}{\overset{H_0}{\gtrless}} \|x-m_0\|^2.$$

$$\|x-m_1\| \underset{H_1}{\overset{H_0}{\lesssim}} \|x-m_0\| \Rightarrow \text{min distance classifier}$$



② $P_0 = P_1$, $\Sigma_0 = \Sigma_1 = \sigma^2 I$

$\|m_0\|^2 = \|m_1\|^2$

$$\Rightarrow m_1^T x \underset{H_0}{\overset{H_1}{\gtrless}} m_0^T x$$

$$\|x-m_1\|^2 = \|x\|^2 - 2m_1^T x + \|m_1\|^2.$$

$\Rightarrow$ correlation classifier

---

② "Soft" classifier

TWO-class

H_0: $x \sim N(m_0, \Sigma_0) \rightarrow P(x|H_0)$

H_1: $x \sim N(m_1, \Sigma_1) \rightarrow P(x|H_1)$

"Soft" classifier $\Rightarrow$

$$P[H_0|x] = \frac{P(x|H_0)P[H_0]}{P(x)}$$

($N(m_0, \Sigma_0)$, P_0)

$$P(x) = P(x|H_0)P[H_0] + P(x|H_1)P[H_1]$$

($N(m_1, \Sigma_1)$)

$$P[H_0|x] = \frac{1}{1 + \frac{P(x|H_1)P_1}{P(x|H_0)P_0}}$$

$$\frac{P(x|H_1)P_1}{P(x|H_0)P_0} \leftarrow \frac{|\Sigma_1|^{-\frac{1}{2}}}{|\Sigma_0|^{-\frac{1}{2}}} e^{-\frac{1}{2}\left(\|x-m_1\|_{\Sigma_1}^2 - \|x-m_0\|^2\right)}$$

quadratic in x.

4

2018年1月29日 18:00

$$\|x-m\|^2 = \|x\|^2 - 2m^T x + \|m\|^2 \Big|_{m_1}^{m_0}$$

Special case.

$P_0 = P_1$

$\Sigma_0 = \Sigma_1 = \sigma^2 I$

$$P[H_0|x] = \frac{1}{1+e^{-\frac{1}{2\sigma^2}\left((2(m_0-m_1)^T x + \|m_1\|^2 + \|m_1\|^2)\cdot(-1)\right)}}$$

$$= \frac{1}{1+e^{-w^T x}} \leftarrow \begin{bmatrix} x \\ 1 \end{bmatrix} \text{ augmented.}$$

Boundary. for general case. Quadratic

$\Rightarrow$ logistic regression $y = f(x) = P[x \in H_0] = P[H_0|x]$

Boundary, $P[H_0|x] = \frac{1}{2}$.

$\Rightarrow$ $w^T x = 0$, linear equation

---

5) optimal Regression w/ Guassian

① optimal Regression

input. $X \longrightarrow N(m_x, \Sigma_{xx})$

output y. $\longrightarrow N(m_y, \Sigma_{yy})$

$$(X,y) \sim N\left(\begin{bmatrix} m_x \\ m_y \end{bmatrix}, \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix}\right)$$

Regression Problem.

Estimate y from x.

optimal estimate $= E[y|x]$

$$= m_y + \Sigma_{yx}\Sigma_{xx}^{-1}(x - m_x)$$

$$= \Sigma_{yx}\Sigma_{xx}^{-1} x + (m_y - \Sigma_{yx}\Sigma_{xx}^{-1} m_x)$$

$$= w^T x \leftarrow \text{Augmented } \begin{bmatrix} x \\ 1 \end{bmatrix}$$

$\Longrightarrow$ linear regression

$w^T \Rightarrow$ matrix when y has multi-D

Data

$x_i \longrightarrow \hat{y}_i = E[y|x_i] = f(x_i)$

5

2018年1月29日 18:00

definition: $E\left[(y-E[y|x])(y-E[y|x])^T\right]$ (est. error)

$\Sigma_{y|x} = \Sigma_{yy} - \Sigma_{yx}\Sigma_{xx}^{-1}\Sigma_{xy}$ $\Rightarrow$ How good the estimation is?

error covariance matrix MSE covariance matrix.

---

6) "Training" for Gaussian models:

① Classification

H_0: $X \sim N(m_0, \Sigma_0)$

H_1: $X \sim N(m_1, \Sigma_1)$

② Regression:

$X \sim N(m_x, \Sigma_{xx})$

$y \in N(m_y, \Sigma_{yy})$

$(x,y) \sim N\left(\begin{bmatrix} m_x \\ m_y \end{bmatrix}, \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix}\right)$

training: parameters estimation ✡

$D \longrightarrow m, \Sigma$

$D = \{x_1, x_2, \cdots x_n\}$

MLE. Gaussian

$(\hat{m}, \hat{\Sigma}) = \arg\max_{m,\Sigma} \log P(D|m,\Sigma)$

$\hat{m} = \frac{1}{n}\sum x_i$ = Sample mean

$\hat{\Sigma} = \frac{1}{n}\sum (x_i - \hat{m})(x_i - \hat{m})^T$.

$P(m,\Sigma)$: Prior

MAP: (max a posteriori) estimate

$\hat{m}, \hat{\Sigma} = \arg\max_{m,\Sigma} \{\log P(m,\Sigma|D)\}$ (Gaussian; Wichout distribution)

$= \arg\max \{\log P(D|m,\Sigma) + \log P(m,\Sigma)\}$

Useful. when the amount of Data is not large.

Clustering

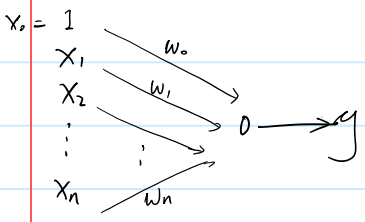
2018年1月29日 18:00

3. Neural Networks

1) Neural networks

① a "neuron"

linear function + a non-linearity.



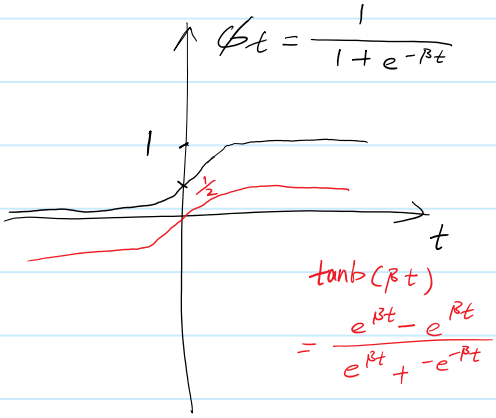
$$y = \phi\left(\sum w_i x_i\right)$$

linear

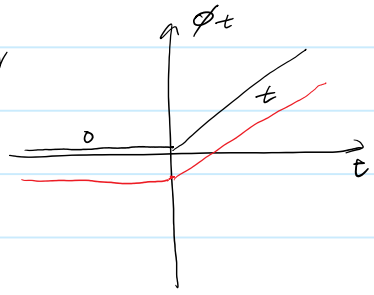
ϕ: nonlinear

why ?

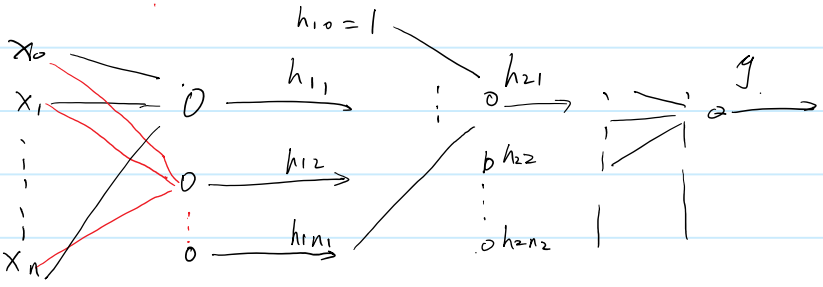
$\sum w_i x_i$: Projection (feature extraction, compression) from a vector to single variable.



Rect linear
Relu



② A neural network.



$$x \xrightarrow{w_1} h_1 \xrightarrow{w_2} h_2 \cdots \rightarrow y$$

$$\mathbb{R}^n \quad \mathbb{R}^{n_1} \quad \mathbb{R}^{n_2} \rightarrow \mathbb{R}$$

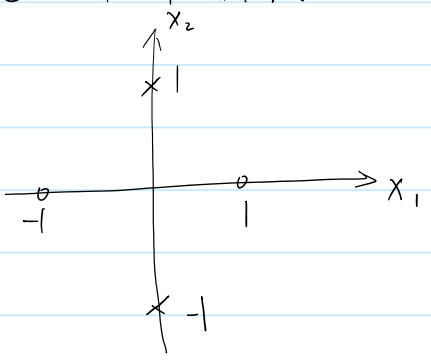
Set up: fully connected feed forward.

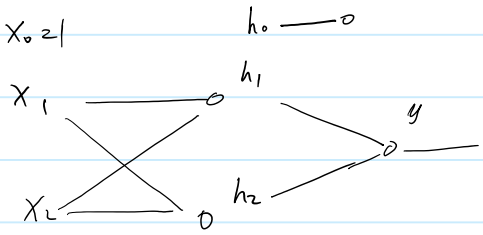
- F.F.
  - MLP
  - CNN
  - AE
- N.N
  - RNN
    - Recurrent (has feedback)

7

2018年1月29日 19:53

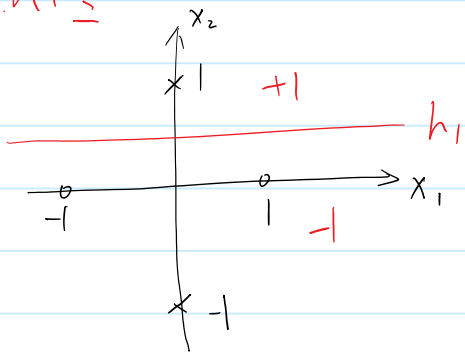
③ XOR Problem.



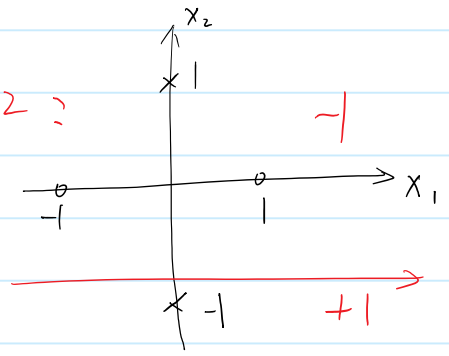


$\mathbb{R}^2 \longrightarrow \mathbb{R}^2$.

h_1 :



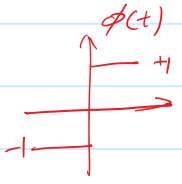
h_2 :



$h_1(x_1, x_2) = \begin{cases} +1, & x_2 > \frac{1}{2} \\ -1, & x_2 < \frac{1}{2} \end{cases}$.

$= \phi(w_1 x_1 + w_2 x_2 + w_0) \Leftarrow$

$w_1 = 0 \quad w_2 = 1 \quad w_0 = -\frac{1}{2}$

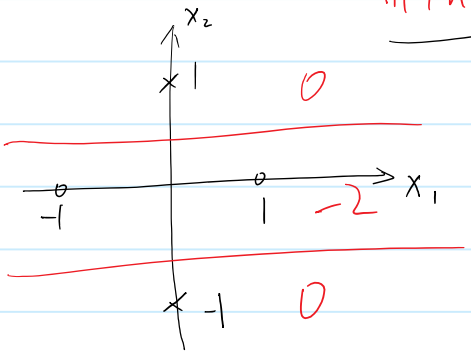


non-linear function

$h_1 + h_2$

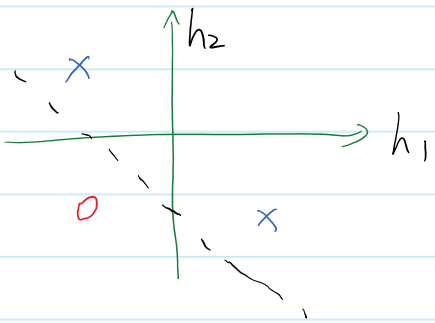
$h_1 + h_2 = y$

linear



before $h_1 + h_2$

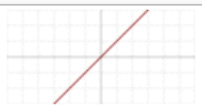
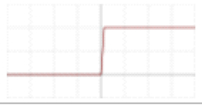
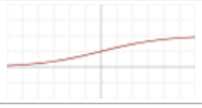
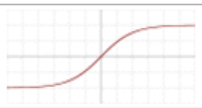
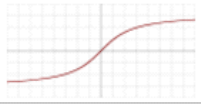
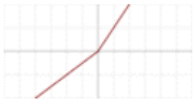
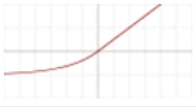
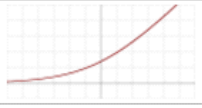
⇓ After



8

Sunday, March 25, 2018 10:55 AM

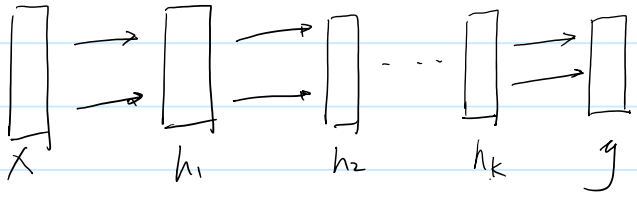
Activation function

Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1-f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2+1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1+e^x)$	$f'(x) = \frac{1}{1+e^{-x}}$

Week3

2018年2月5日 17:08

Review: Neural networks



$$h_k = \phi(W^{k-1} h_{k-1})$$

- ϕ: non-linear, component-wise
- W^{k-1}: matrix
- h_{k-1}: column vector

$$\phi\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} \phi(x_1) \\ \phi(x_2) \\ \vdots \\ \phi(x_n) \end{pmatrix}$$

2) Function Approximation by NN.

① function appr. est. problem.

function: $y = f(x)$

$f(\cdot)$: unknown

Have: $D = \{(x_i, y_i)\}$

Function est./appr. $\hat{f}(\cdot)$.

$D \longrightarrow \hat{f}(\cdot)$

interpolation

regression

①

Example: LR:

$\hat{f}(x) = w^T x$ = linear function.

(n dim; # of terms O(n))

$$w = E[xx^T]^{-1} E[xy]$$

$$\approx \left(\sum \frac{x_i x_i^T}{n}\right)^{-1} \approx \frac{\sum x_i y_i}{n}$$

Adv.

Simple

Scales well w/ dim

Prob:

Large error

when $f(\cdot)$ is nonlinear.

2018年2月5日 17:09

② Example: polynomial problem.

1D: $\hat{f}(x) = a_0 x^k + a_1 x^{k-1} + \cdots + a_k$

$$(a_0^*, a_1^*, \cdots a_k^*) = \underset{a_0, \cdots a_k}{\arg\min} \; E[(y - \hat{f}(x))^2]$$

Adv: simple; nonlinear for 1D problem

Problem: doesn't scale well for n-dim.

2D polynomial of order k ($k=2$)

$$\hat{f}(x) = ax_1^2 + bx_2^2 + cx_1 + dx_2 + ex_1x_2 + d$$

$\alpha_0=0, \alpha_1=2, \alpha_2=0$

$\alpha_0=1, \alpha_1=1, \alpha_2=0$

$$\hat{f}(x) = ax_1^2 + bx_2^2 + cx_1 + dx_2 + ex_1x_2 + d$$

$\alpha_0=0$, $\alpha_1=1$, $\alpha_2=1$

$\alpha_0\geq 2$, $\alpha_1=0$, $\alpha_2=0$

n-dim polynomial order k.

$$\hat{f}(x) = \sum x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n} + c$$

$\alpha_0 + \alpha_1, \alpha_2 \cdots \alpha_n$.

$\alpha_0 + \alpha_1 + \alpha_2 + \cdots + \alpha_n = k$.

$\alpha_0 \geq 0, \alpha_1 \geq 0, \alpha_2 \geq 0 \cdots \alpha_n \geq 0$

$C_{m_1}^{m_2}$

Bars & Stars

# of terms $O(n^k)$

$n = 256 \times 256$, $k = 10$ $\Rightarrow (2^{16})^{10} = 2^{160}$

③ Example: Fourier transform.

1D: $\hat{f}(x) = \sum_{m=0}^{M-1} f_m e^{j w_0 m x}$ # of terms: $O(M)$.

2D: $f(x) = \sum_{m_1, m_2} f_{m_1, m_2} e^{j w_0 (m_1 x_1 + m_2 x_2)}$ # of terms: $O(M^2)$

⋮

nD: # of terms $O(M^n)$

Adv. Fourier can represent any formulation

Problem: only in low Dim problem

2018年2月5日 17:09

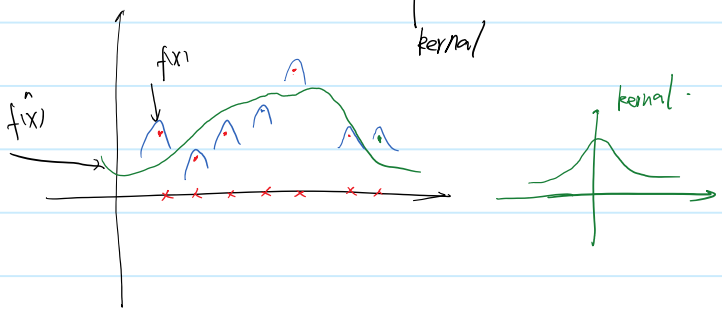
④ Example, kernal regression (sum)

$$\hat{f}(x) = \sum w_i \phi(x - x_i)$$

kernal

need training data every time.

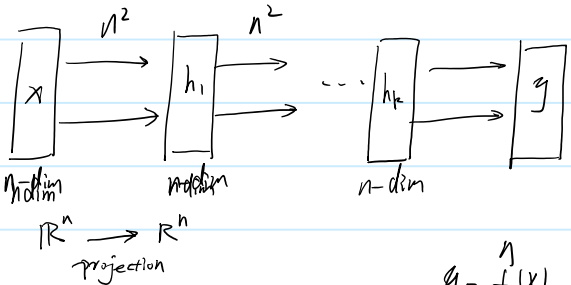
x_i



n-dim, 100^n point.

$n = 100 \longrightarrow 100^{100}$ point.

⑤ Example, NN



$g = \hat{f}(x) = w^k h_k(x)$

$h_1 = \phi(w^1 x)$

n dim column vector

n-dim column vector

n×1 matrix

$O(n^2)$

orthonormal

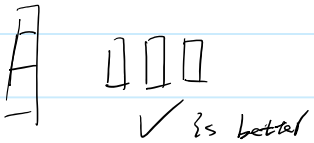
Total # of weights.

NN's ability to approximate

→ only one hidden layer is sufficient

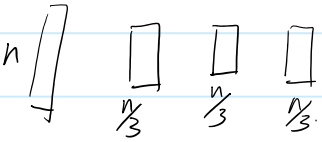
(may need a expotential # of nodes)

For same nodes,



✓ is better

n input, n hidden layer nodes



$(\frac{n}{3} \times n) + (\frac{n}{3})^2 + (\frac{n}{3})^2$

$= \frac{n^2}{3} + \frac{n^2}{9} + \frac{n^2}{9} = \frac{5}{9}n^2$

2018年2月5日 17:09

3) Nerual Network trainning

① problem:

$D \longrightarrow w$

② Optimization

$$w^{*} = \arg\min_{w} J(w)$$

example: MSE

$$J(w) = E\left[\left(y - \hat{f}(x)\right)^2\right]$$

output, NN, input to NN

$$= E\left[(y - \hat{f}(x/w)^2\right]$$

$$= \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{f}(x_i/w)\right)^2.$$

example: Cross entropy

random variable X, X discrete.

X	x_1, x_2 ... x_n
P	P_1, P_2, ... P_n

$\Rightarrow$ true PD (Probability Density)

Approximation to P, q

X	x_1, ... x_n
q	$q(x_1)$, ... $q(x_n)$

in nn, $q = q(x) = \hat{f}(x/w)$

$$H(P,q) \approx \sum -\left(y_i \log \hat{f}(x_i) w) + (1-y_i)\log(1-\hat{f}(x_i/w)\right)$$

binary case

$$H(P,q) = E_P[-\log q(x)]$$

$$= -\sum P_k \log q(x_k)$$

$$= -\sum P_k \log q_k$$

$\Leftarrow$ $H(P,q) = H(P) + D(P\,//\,q)$

$H(P)$: entropy of X, $-\sum P_k \log P_k$

$D(P\,//\,q)$: KL distance from P to q, $\sum P_k \log \frac{P_k}{q_k}$

$D(P\,//\,q) \geq 0$, $= 0$ when $P = q$

5

2018年2月5日 17:09

Example: likelihood function.

$$w = \arg\min_w E[-\log q(y|x,w)]$$

q: conditional pdf of y given x.

w: parameters

$$E[-\log q(y|x,w)]$$

$$\approx \frac{1}{N}\sum_{i=1}^{N}(-\log q(y_i|x_i,w))$$

Example: $q(y|x,w)$

$$y|x,w \sim N(m,\sigma^2)$$

$$m = \hat{f}(x|w)$$

($\hat{f}$: neural network)

$$\Rightarrow E[-\log q(y|x,w)]$$

$$= c + E\left[\frac{(y-\hat{f}(x|w))^2}{2\sigma^2}\right]$$

$$= MSE$$

Example: same as before.

$$m = \hat{f}(x|w) = \text{a NN}$$

$$\sigma^2 = \hat{g}(x|w') = \text{another NN}$$

$$E[-\log q(y|x,w,w')] = c + E[\log g(x|w')] + E\left(\frac{(y-\hat{f}(x|w))^2}{2\hat{g}(x|w')}\right)$$

$$\approx c + \frac{1}{N}\sum\left(\log \hat{g}(x|w') + \frac{(y_i - f(x_i|w))^2}{2\hat{g}(x_i|w')}\right)$$

→ Criterion

→ optimization.

2018年2月5日 17:09

② Optimization.

→ traditional opt. technique (e.g. gradient descent)

→ problems specific to deep structure (nested structure)

→ fitting the training data. VS. generalization.

overfit ⇒ regularization

⇒ robust estimation

Week 5

2018年2月19日 17:09

3) (Deep Neural network training)

① General problem

② Criterion

4) Review of optimization

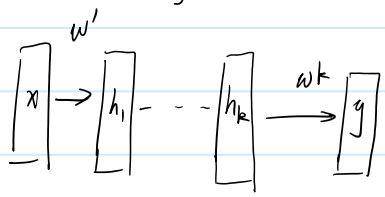
5) Issues related to Deep neural network.

---

① General problem

Training data $D = \{(x_i, y_i)\}$

Neural Network



Parameterize by w

$y = f(x|w)$

Weight $[w^1, w^2 \cdots w^k] = w$

Training $D \rightarrow w$

Solved as a optimization problem.

$w^* = \underset{w}{\operatorname{argmin}}\ J(D|w)$

② Examples of $J(D|w)$

training data

$E[J(x,y)|w]$

example: MSE

$J(X, Y|w) \rightarrow E[(Y - f(x|w))^2]$

$\approx c\sum (y_i - f(x_i|w))^2$.

2018年2月19日 17:39

Example: Cross Entropy

$$J(x, y|w) = E[-\log f(x|w)]$$

true $P[Y=1]$ → y; neural network output $= P[Y=1]$ (estimated) → $f(x|w)$

$$= -P(x)\log f(x|w) - (1-P(x))\log(1-f(x|w))$$

$$\approx c\sum[-y_i \log f(x_i|w) - (1-y_i)\log(1-f(x_i|w))]$$

Example: (conditional) likelihood function

Data. $P(x)$, pdf of rv x

$$P(x) = P(x|\theta)$$ — parameters.

likelihood function.

$D = \{x_1, x_2 \cdots x_n\}$ data, iid samples from x.

likelihood function, $P(D|\theta) = P(x_1, x_2 \cdots x_n|\theta)$

Example: $P(x|\theta)$ = Gaussian w (m, σ^2)

$\theta = (m, \sigma^2)$

$$P(D|\theta) = P(x_1 \cdots x_n|\theta)$$

$$\pi(P(x_i|\theta) = \pi \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x_i-m)^2}{2\sigma^2}}$$

MLE, $\hat{\theta} = \arg\max_{\theta} \log P(D|\theta)$.

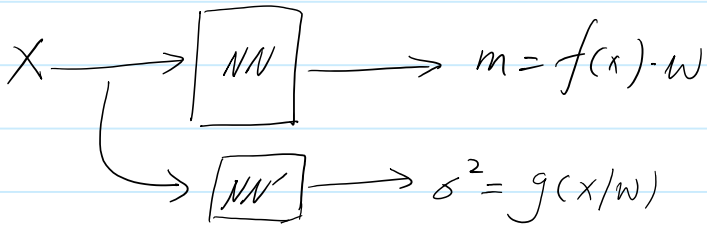
Gaussian example,

$$\hat{m} = \frac{1}{n}\sum x_i$$

3

2018年2月19日 17:40

Applied to Nerual Network



$m = f(x/w)$

likelihood function $P(y|x,\theta)$ = Gaussian (m, σ^2) (m: unknown)

any Density function $= P(y|x,m) = P(y|x, f(x/w))$

NN training as MLE

likelihood function $P(y_1 \cdots y_n | x_1 \cdots x_n, m)$

$$= \pi_i \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y_i - f(x/w))^2}{2\sigma^2}}$$

MLE, $w^* = \arg\max_w P(y_1, y_2, \cdots y_n | x_1 \cdots x_n, m)$

= non-linear function of weights.

(Same as MSE)

MLE ⇒ ① enlarge the training criteria.

ex. σ^2 (unknown), m → know

② generte powerful Density function

4

2018年2月19日 17:40

4) Review. optimization methods.

convex → unique global min

non convex → many local min

⇒ saddle points

neural network

constrained < equality / inequality — KKT conditions

unconstrained

$\nabla f = 0$

$\nabla^2 f > 0$ — positive definite.

① simplest case

$$X^* = \arg\min_x f(x)$$

necessary condition, not sufficient $\nabla f(x) = 0$

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}$$

Problem: $\nabla f(x) = 0$ often can not be solved analytically.

Solution: iterative solution

② iterative solutions:

Gradient descent (GD)

Newton method → quasi newton → low memory quasi newton

Second order matrix

nxn matrix

Boyd "convex optimization"

conjugate gradient.

$Ax = b$

$\min_x \left(\frac{1}{2} x^T A x - x^T b\right)$

$\downarrow \nabla$

$Ax - b = 0$

momentum method → parameters set by hand.

5

2018年2月19日 17:39

③ newton method.

$x_0 \to x_1 \to x_2 \cdots \to x^*$

(recall $x^* = \min_x f(x)$)

$\nabla f(x) = 0$

$g(x) = \nabla f(x)$

Given x_n

Find x_{n+1}

$g(x_{n+1})$ ideally $g(x_{n+1}) = 0$

Tylor expansion 0TH

$$g(x_{n+1}) \approx g(x_n) + \underbrace{\left(\nabla g(x_n)\right)}_{\text{matrix}} \underbrace{(x_{n+1} - x_n)}_{\text{vector}}$$

$\| \quad 0$

$$g(x_n) + \nabla g(x_n)(x_{n+1} - x_n) = 0$$

$$x_{n+1} - x_n = -\left(\nabla g(x_n)\right)^{-1} g(x_n)$$

$$= -\underbrace{\left(\nabla^2 f(x)\right)^{-1}}_{\text{Hessian matrix}} \left(\nabla f(x_n)\right)$$

actual algorithm. (line search)

$$x_{n+1} = x_n - \alpha \left(\nabla^2 f(x_n)\right)^{-1} \left(\nabla f(x_n)\right)$$

$$\nabla^2 f(x_n) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial f}{\partial x_1 x_2} & \cdots & \frac{\partial f}{\partial x_1 x_n} \\ & & & \\ & & & \frac{\partial f}{\partial x_n x_n} \end{bmatrix} \to \frac{\partial^2 f}{\partial x_i \partial x_j}$$

Problems: dim is high

Solutions → quasi, low-mem Newton method.

→ drastic approximation

$(\nabla^2 f)^{-1} \approx I$ $\Rightarrow x_{n+1} = x_n - \alpha \nabla f(x_n) \Rightarrow$ GD

2018年2月19日 17:39

④ Gradient Descent

learning rate
step size.

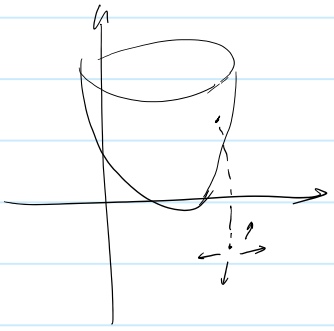
$$x_{n+1} = x_n - \alpha \nabla f(x_n)$$

direction fast descent.

$-\nabla f(x_n)$

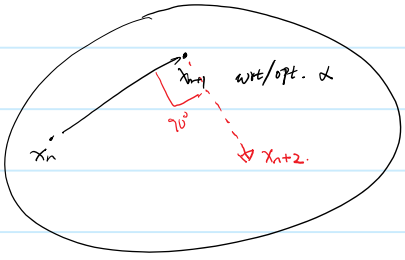
① fast change $\nabla f(x)$
$-\nabla f(x)$

② $-\nabla f(x)$ descent.



line search,

Select α, $f(x_{n+1}) = f(x_{n+1}(\alpha))$. is min wrt α.

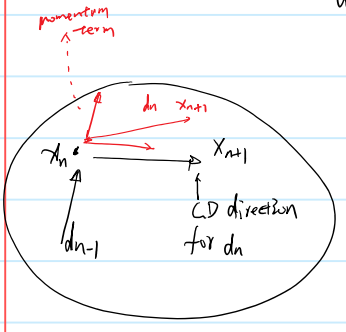


---

⑤ Conjugate gradients method.

Basic Idea.

d_{n-1} direction generated x_n.



$$x_n = x_{n-1} + \alpha d_{n-1}$$

$$d_n = -\nabla f(x_n) + \beta_n d_{n-1}$$

CD: $d_{n-1} = -\nabla f(x_{n-1})$

orthogonal

if $u \perp v$ then $u^T v = 0$, $u \cdot v = 0$

conjugate: $u^T A v = 0$

A-orthogonal

A: spp matrix

symmetric Define positive.

7

2018年2月19日 17:39

$d_n^T \nabla^2 f(x_n)\, d_{n-1} = 0$ Conjugate relation

$$\beta_n = \frac{(\nabla f(x_n) - \nabla f(x_{n-1}))^T \nabla f(x_{n-1})}{\nabla f(x_{n-1})^T \nabla f(x_{n-1})}$$

G. Strang, Applied math.

?, numerical Recipe. in C

optimal point x^*

Δ $d_k A d_{k-1} = 0$ $\dim(x) = n$.

$d_i^T A d_j = 0$ $i \neq j$

n d_k's. ; linearly independent.

$x^* = \sum \alpha_i d_i$ $\Rightarrow$ $d_1 \to d_2 \to d_3 \cdots d_n$.

α_2 α_3 α_n.

$d_k^T A x^* = d_k^T A (\sum \alpha_i d_i)$

$Ax = b$, $Ax^* = b$ $\Longrightarrow$ Δ $d_k^T b = \alpha_k d_k^T A d_k$

$\alpha_k =$

---

5) Training Deep Neural Network.

① Gradient calculation. : BP

Δ ② Vanishing/Exploding gradients : BN

Bech normalization

$x \to \square \to \square \cdots \square \to y$

$y = (\prod w^k)\, x$.

③ generalization BN.

Week6

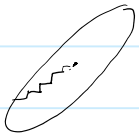
2018年2月26日 17:39

Review: Optimization methods.

Opt. problem: $X^* = \arg\min_x f(x)$

GD: $X_{n+1} = X_n - \varepsilon \nabla f(x)\big|_{x=x_n}$.

$\nabla f(X_n)$



Newton: $X_{n+1} = X_n - \varepsilon (\nabla^2 f(x_n))^{-1} \nabla f(x_n)$

$(\dim(x))^2$ in complexity

→ quasi newton

Conjugate gradient $d_n = -\nabla f(x_n) + \beta_n d_{n-1}$

$X_{n+1} = X_n + \varepsilon d_n$.

---

5) Training Deep Neural Network.

① Stochastic GD → SGD

② Several Neural Network problems.

Vanishing / exploding gradient: BN (Batch normalization)

SGD being too slow — momentum

— adaptive learning rate

Generalization — regularization — drop out

— L1, L2 regularization

③ Gradient Calculation: BP.

2018年2月26日 16:53

① Stochastic GD: $h(x) = h(x|w)$

$$w^* = \min_w f(w), \quad f(x) = E\left[(y - h(x|w))^2\right]$$

Ture GD:

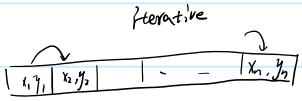
$$w_{n+1} = w_n - \varepsilon \nabla_w E\left[(y - h(x|w))^2\right]$$

Batch method:

$$w_{n+1} = w_n - \varepsilon \nabla_w E\left[(y - h(x|w))^2\right]$$

$$\approx w_n - \varepsilon \nabla_w \left(\frac{\sum (y_i - h(x_i|w))^2}{n} \right)$$

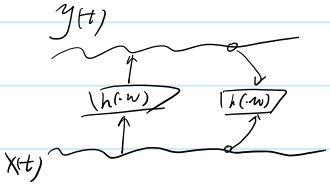
Iterative



$\{(x_i, y_i)\}$: training data.

SGD: $w_{n+1} = w_n - \varepsilon \nabla_w (y_j - h(x_j|w))^2$ / Randomly choose one data point. at one time.

LMS/adaptive filtering



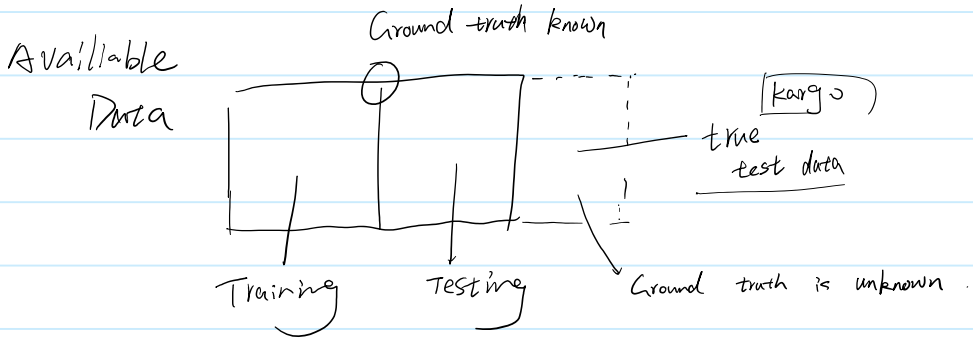
most used mini batch SGD, random subset / specify the size of mini batch.

$$w_{n+1} = w_n - \varepsilon \nabla_w \frac{\sum_{k \in D_k} (y_{ik} - h(x_{ik}|w))^2}{k} \quad (w = w_n)$$

D_k = random subset of D w/ k samples.

3

2018年2月26日 17:09



---

② Several neural network (Deep) problems.

— Proposed solutions.

<1> SGD too slow reason: GD is too slow

Proposed solution: momentum method

m_n = new direction for generating w_{n+1}

$w_{n+1} = w_n + \varepsilon m_n$.

$m_n = \beta m_{n-1} - \alpha \nabla J(w)|_{w=w_n}$

$J(w) = E[(y - h(x|w))^2]$

$\approx \frac{1}{k} \sum (y_{ik} - h(x_{ik}|w))^2$.

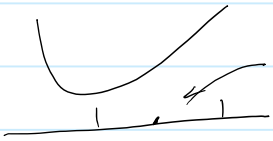
Conjugate gradient descent,

$d_n = \beta_n d_{n-1} - \alpha \nabla J(w)$

$w_{n+1} = w_n + d_n$.

4

2018年2月26日 16:54



Proposed solution. increasing learning rate

1st step. detect flat region.

$|\nabla f(w)|$ too small

$$w_{n+1} = w_n - \alpha \frac{\nabla_w J(w_n)}{\sqrt{s_n + \varepsilon}}$$

point wise

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix} / \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/2 \end{bmatrix}$$

$\Rightarrow$ RMS prop

s_n becomes small when $\nabla J(w_n)$

RMS Prop + Momentum $\longrightarrow$ Adam

$$s_n = \underbrace{\beta s_{n-1}}_{\text{LPF}} + (1-\beta) \nabla J(w_n) \cdot \nabla J(w) \quad \beta \in (0,1)$$

point wise

$\Rightarrow \begin{bmatrix} 1 \\ 2 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 3 \\ 8 \end{bmatrix}$

<2> Vanishing / Exploding Gradient.

$\left(\prod_{m=1}^{M} w_m \right)$ M is large.

$w_m > 0$

$w_m > 1 \Rightarrow$ prod $\to +\infty$

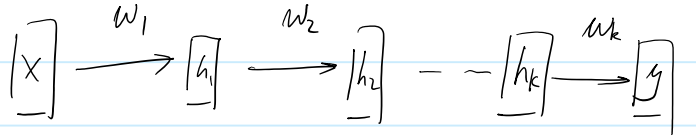
$w_m < 1 \Rightarrow$ prod $\to 0$

finite produce, most $w_m = 1$

5

2018年2月26日 16:54

Neural networks.



normalize $w_k \cdot h_k$ around 1

$h_1 = \phi(w_1, x)$

$h_2 = \phi(w_2, h_1) = \phi(w_2 : \phi(w_1, x))$

$\vdots$

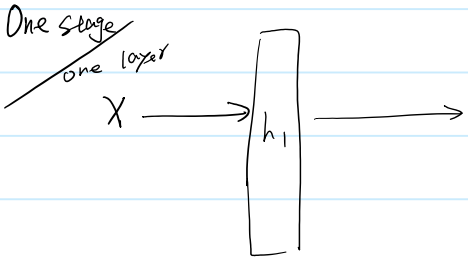
$y = \phi(w_k, h_k) = \phi(\cdots \phi(w_k, h_k))$

$\phi = I \Rightarrow y = (\prod w_k) x$

↳ linear case.

Batch normalization

One stage / one layer



$h_1 = \phi(w_1 x)$

$= \phi(u) \quad u = w_1 \cdot x$

Want u : to have a $N(0,1)$

Gaussian $\begin{bmatrix} E[u] \\ Var[u] \end{bmatrix}$

r.v. $u \longrightarrow N(0,1)$

$t = \dfrac{u_i - E[u]}{Var[u]}$ ← constant; calculate using mini-batch; constant.

$\Downarrow$

$t \sim N(0,1)$ $\begin{cases} E[t] = 0 \\ Var[t] = 1 \end{cases}$

replace $h_1 \phi(w_1 x) = \phi(u)$

by $h_1 = \phi(\alpha t + \beta)$

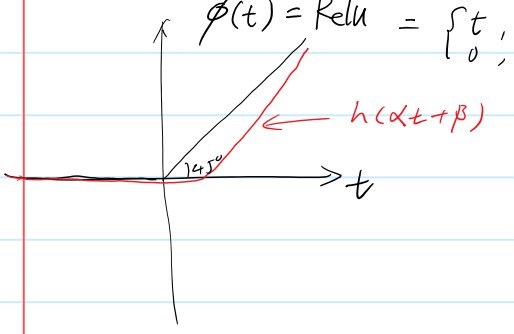
added parameter to be trained.

2018年2月26日 19:14

$t \sim N(0,1)$

$\alpha t + \beta \sim N(\beta, \alpha^2)$

$\phi(t) = \text{Relu} = \begin{cases} t, & t>0 \\ 0, & t<0 \end{cases}$



w/o normalization $\alpha, \beta \rightarrow w_k$.

w normalization need α, β

At step n

α_n, β_n w_{1n} → t_n (through normalization)

$\Rightarrow h_1 = \phi(\alpha_n t_n + \beta_n)$

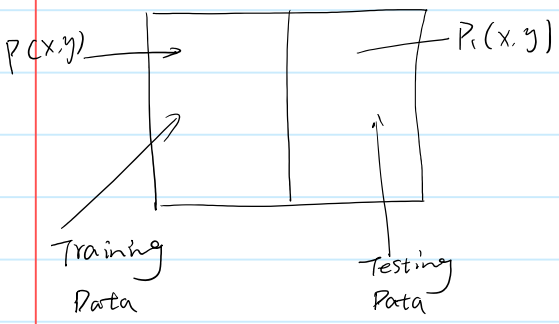
→ Calculate Gradient wrt w_n, α_n, β_n

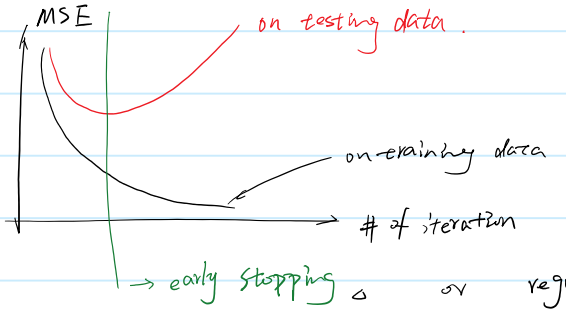
→ $\alpha_{n+1}, \beta_{n+1}, w_{1,n+1}$

→ Step n+1

---

(5) Generalization.





$MSE = E[(y - h(x|w))^2]$

On training data.

$= E_p[(y - h(x|w))^2] \approx \frac{1}{n} \sum (y_i - h(x_i|w))^2$.

Different Density P

On testing data

$= E_{p_1}[(y - h(x|w))^2] \approx \frac{1}{n} \sum (y_j - h(x_j|w))^2$.

7

2018年2月26日 19:14

Proposed solution ① early stopping
② regularization

example: 10,000 samples.
10^6 weights

under constrained problem.
→ introduce conserant, most weights equals to zero

10 non zero weights
→ $C_{10,000}^{10}$ ⇒ too many comb. to check.

Solution: regularization.
L_1 regularization.

$\nabla J(w) = E[(y - h(x|w))^2]$

→ $\nabla J(w) + \lambda \|w\|_1$

$\begin{bmatrix} 1 \\ 2 \\ \vdots \\ 10 \end{bmatrix}$ ⇒ non-sparse

$\begin{bmatrix} 1 \\ 2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$ ⇒ Sparse most are 0

$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$ $\|x_1\|_1 = \sum_{i=1}^{m} |x_i|$.

⇒ Sparse solution

L_2:
→ $\nabla J(w) + \lambda \|w\|_2$

$\|x\|_2 = \sum x_i^2$. $x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$

⇒ Small weights

Week7

2018年3月5日 19:14

Training Deep Neural Networks

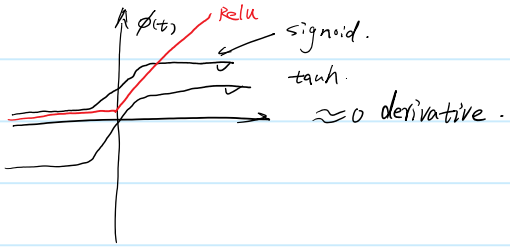
— Calculate gradient, BP

— exploding / vanishing gradient, Batch normalization.

selecting initial value — He initialization

↑

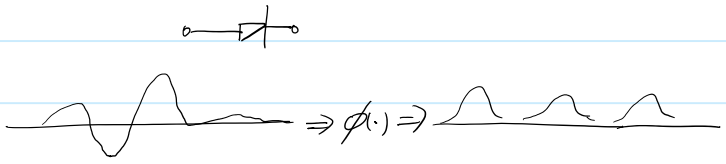
Relu activation



Relu

$$\phi(t)=\begin{cases}0, & t<0\\ t, & t\geq 0\end{cases}$$

Rectification



$\Rightarrow \phi(\cdot) \Rightarrow$

$\phi(\alpha t+\beta)$

normalized $N(0,1)$

— GD too slow

— momentum similar to CGM

— adaptive learning rate, ∇J $\odot$ ∇J point by point production

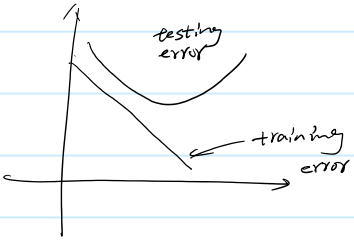
} ⇒ Adam.

∇J: optimization

— Regularization.

training error, already small & decrease

testing error, & increase



2018年3月5日 19:14

l_2, l_1 regularization

early stop

drop off / randomly drop off a set of nodes.

rate = $\frac{1}{2}$.

train the rest

Gradient calculation

Review: the chain rule.

$f(x) = y$ $\quad \frac{dy}{dt} = \frac{dy}{dx} \cdot \frac{dx}{dt} = f' \cdot g'$

$x = g(t)$

$$t \xrightarrow{g} x \xrightarrow{f} y$$

$$(t,s) \xrightarrow{g} (x,y) \xrightarrow{f} (u,v)$$

$$\frac{\partial u}{\partial t} = \frac{\partial u}{\partial x} \cdot \frac{\partial x}{\partial t} + \frac{\partial u}{\partial y} \cdot \frac{\partial y}{\partial t}$$

$$(x_1 \cdots x_N) \longrightarrow (v_1 \cdots v_I) \longrightarrow (u_1 \cdots u_J) \longrightarrow (y_1 \cdots y_K)$$

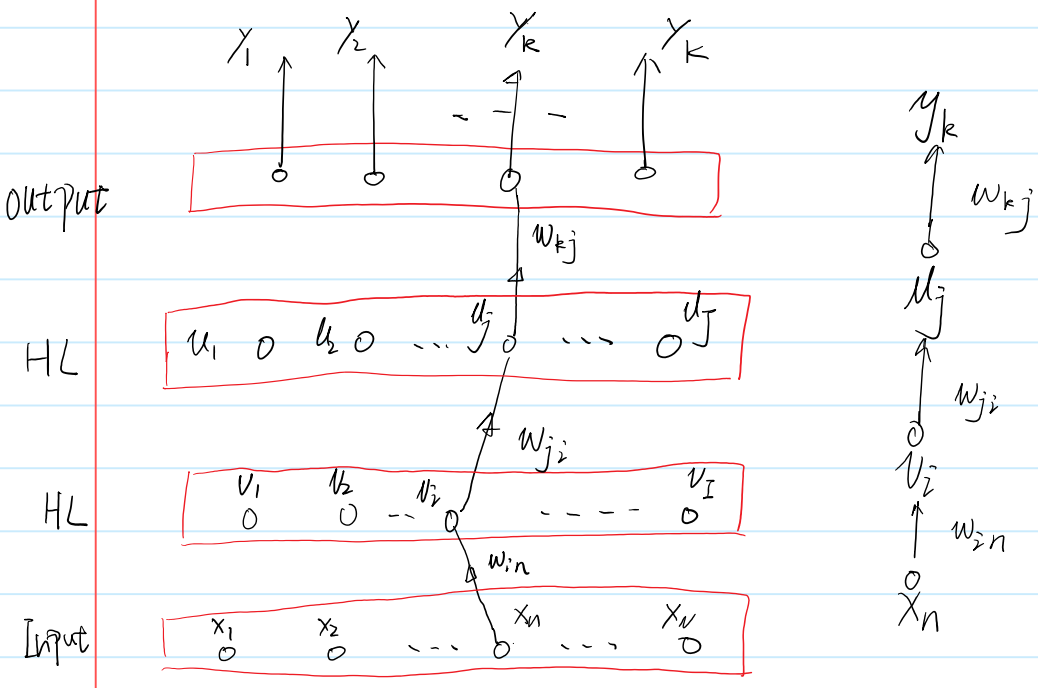
input HD1 HD2 output

$$\frac{\partial y_k}{\partial v_i} = \sum_j \frac{\partial y_k}{\partial u_j} \cdot \frac{\partial u_j}{\partial v_i}$$

$$\frac{\partial y_k}{\partial x_n} = \sum_{i,j} \frac{\partial y_k}{\partial v_j} \cdot \frac{\partial u_j}{\partial v_i} \cdot \frac{\partial v_i}{\partial x_n}$$

3

2018年3月5日 15:37



Training

$$J(w) = \sum \frac{1}{2} \|y - t\|^2$$

(t, x): target output, input

$y = y(x, w)$

$$J(w) = \sum \frac{1}{2} \|NN(x_i, w) - y_i\|^2$$

NN: neural network

$D = \{(x_i, y_i)\}$

Gradient calculation:

$$\frac{\partial J}{\partial w} = \{\text{all partials of } J \text{ wrt weight}\}$$

tensor

4

2018年3月5日 15:37

$J(w) = \frac{1}{2}\|y(x) - t\|^2$ one sample mini batch.

$= \frac{1}{2}\sum (y_k(x) - t_k)^2$ $\quad y_k(x) = y_k(x, w)$.

$y_k(x)$: variable. x: input sample. t_k: target value.

$= \frac{1}{2}\sum (y_k(x, w) - t_k)^2$.

$\frac{\partial J}{\partial w_{kj}} = \frac{\partial}{\partial w_{kj}} \frac{1}{2}\sum (y_{k'} - t_{k'})^2$ ($y_{k'}$: variable, $t_{k'}$: target value.)

$= \underbrace{\sum_{k'} (y_{k'} - t_{k'})}_{\text{error } \delta_{k'}} \cdot \frac{\partial}{\partial w_{kj}} (y_{k'} - t_{k'})$

$= \sum_{k'} \delta_{k'} \frac{\partial y_{k'}}{\partial w_{kj}} = \delta_k \frac{\partial y_k}{\partial w_{kj}} = \delta_k \phi'(\sum_j w_{kj} \cdot u_j) \cdot u_j$

$y_k = \phi(\sum_j w_{kj'} \cdot u_{j'})$

$\frac{\partial y_k}{\partial w_{kj}} = \phi'(\sum_{j'} w_{kj'} \cdot u_{j'}) \cdot \frac{\partial}{\partial w_{kj}} (\sum_{j'} w_{kj'} \cdot u_{j'})$

$= \phi'(\quad) u_j$

when ϕ is linear $\phi'(\,) = 1$

$\frac{\partial J}{\partial w_{kj}} = \delta_k u_j$

$\frac{\partial J}{\partial w_{ji}} = \frac{\partial}{\partial w_{ji}} \times \frac{1}{2}\sum (y_k - t_k)^2$

$= \sum (y_k - t_k) \frac{\partial y_k}{\partial w_{ji}}$

$= \sum_k \delta_k \frac{\partial y_k}{\partial w_{ji}}$

$\frac{\partial y_k}{\partial w_{ji}} = \frac{\partial y_k}{\partial u_j} \cdot \frac{\partial u_j}{\partial w_{ji}}$

$y_k = \phi(\sum_j w_{kj'} \cdot u_{j'})$

$\frac{y_k}{u_j} = \phi'(\sum_j w_{kj'} \cdot u_{j'})$

$= \phi'(\,) \cdot w_{kj}$

5

2018年3月5日 15:37

$$\frac{\partial u_j}{\partial w_{ji}} = \frac{\partial}{\partial w_{ji}} \phi\left(\sum_{i'} w_{ji'} v_{i'}\right)$$

$$= \phi\left(\sum_{i'} w_{ji'} v_{i'}\right) \cdot v_i$$

$$\frac{\partial J}{\partial w_{ji}} = \sum \delta_k \frac{\partial y_k}{\partial w_{ji}}$$

$$= \sum_k \delta_k \phi'\left(\sum_{j'} w_{kj'} u_{j'}\right) \cdot w_{kj} \cdot \phi'\left(\sum_{i'} w_{ji'} v_{i'}\right) v_i.$$

$$= \left(\sum_k \delta_k \phi'\left(\sum_{j'} w_{kj'} u_{j'}\right) \cdot w_{kj}\right) \cdot \phi'\left(\sum_{i'} w_{ji'} v_{i'}\right) v_i.$$

linear combination error at layer above.

error back-propogation δ_j.

$$= \delta_j \cdot \phi'\left(\sum_{i'} w_{ji'} v_{i'}\right) v_i. \quad \Rightarrow \quad \boxed{\frac{\partial J}{\partial w_{ji}} = \delta_j \phi' v_i}$$

u_j'

$$\frac{\partial J}{w_{kj}} = \delta_k y_k' u_j \qquad \delta_j = \sum_k \delta_k y_k' w_{kj}$$

$$\frac{\partial J}{w_{ji}} = \delta_j u_j' v_i \qquad \delta_j = \sum_k y_k' w_{kj} \delta_k.$$

$$\frac{\partial J}{\partial w_{in}} = \delta_i v_i' x_n. \qquad \delta_i = \sum \delta_j y_j w_{ji}$$

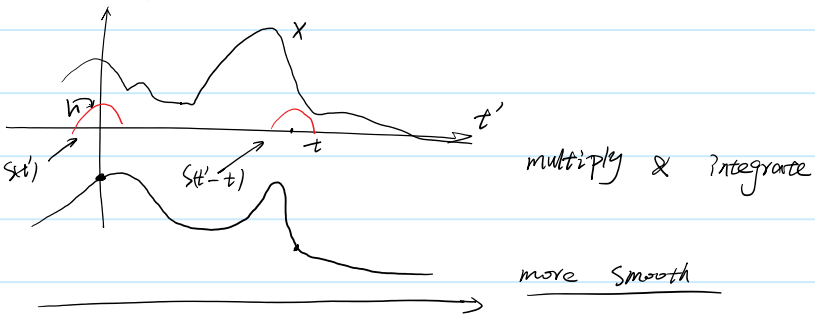
6

2018年3月5日 15:37

4. Convolutional Neural Networks (CNNs)

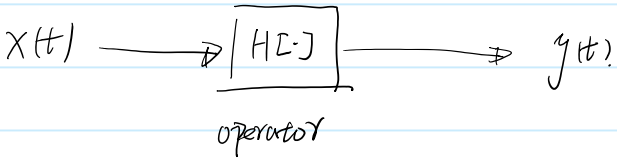
1) Convolution.

"moving average"



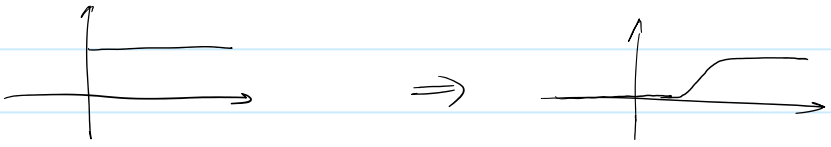
① linear time invariant systems (LSI).

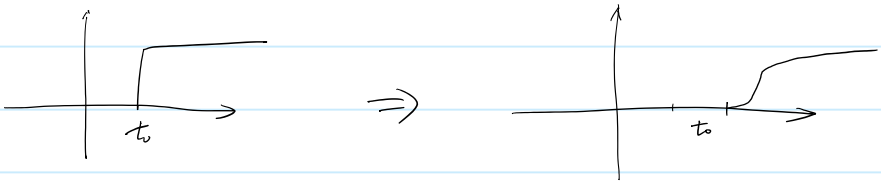
(shift)



linear: $H[aX_1(t) + bX_2(t)] = aH[X_1(t)] + bH[X_2(t)]$

shift invariant:





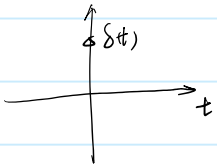
② convolution (LSI system)

$y(t) = (h * x)(t)$

$= \int_{-\infty}^{+\infty} x(t')\, h(t-t')\, dt'$

$= \int_{-\infty}^{+\infty} (x*t-t')\, h(t')\, dt'$

H[·]: LSI complete det. by $H[\delta(t)]$



$\delta(t) = 0, \quad t \neq 0$

$\int \delta(t)\, dt = 1$

$H[\delta(t)] = h(t)$: impulse response

7

2018年3月5日 15:38

flip and move

$$y(t) = \int_{-\infty}^{+\infty} x(t')\, h(t - t')\, dt'$$

$$= \int_{-\infty}^{+\infty} x(t - t')\, h(t')\, dt'$$

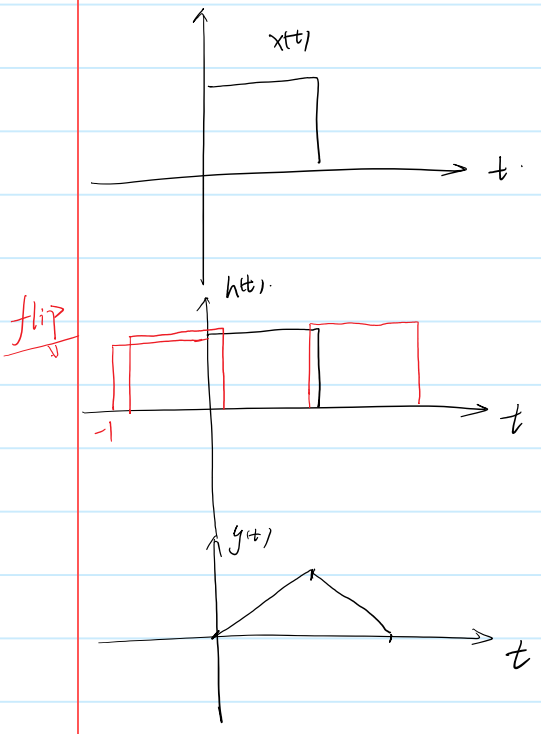
$x(t) \longrightarrow X(\omega)$

$h(t) \longrightarrow H(\omega)$

$y(t) \longrightarrow Y(\omega)$

$Y(\omega) = H(\omega)\, X(\omega)$.

filter, $h(t)$, $H(\omega)$



support $(x(t)) = 1$

$(h(t)) = 1$

{ the length of interval on which it is not zero

$y(t) = (x * h)(t)$

support $(y(t)) = 1 + 1 = 2$.

support $(h_1 * h_2 * h_3 * h_4) = \sum_{i=1}$ support$_i$

Week8

2018年3月12日 15:38

Review

convolution, moving average.

$x(t)$, $h(t)$ → ⊛ ⇒ $y(t)$ $\quad (x*h)(t) = \int_{-\infty}^{+\infty} x(t')\, h(t-t')\, dt'$

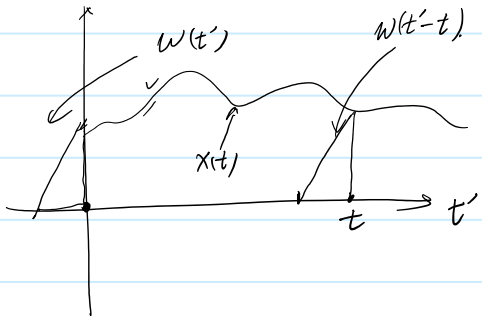
or

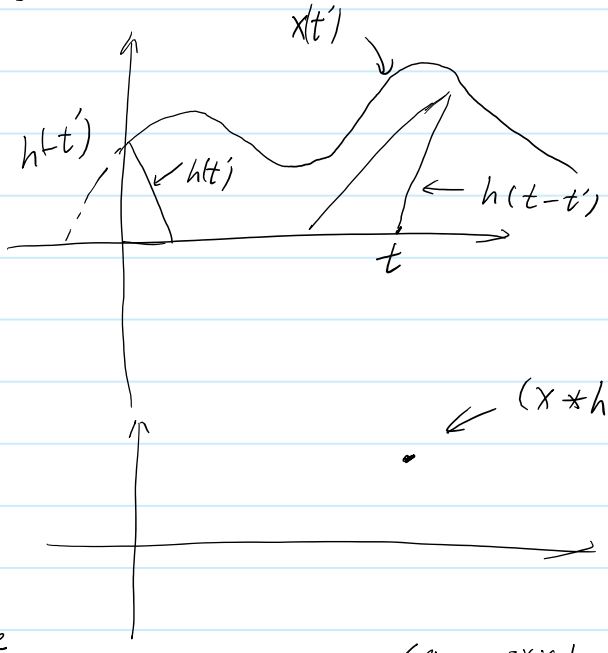
$$\int_{-\infty}^{+\infty} x(t-t')\, h(t')\, dt'$$

moving average.

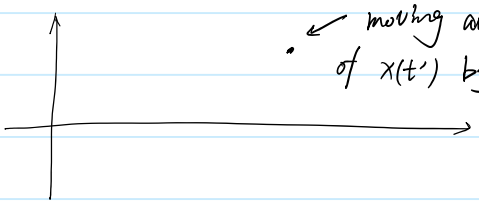
y $x(t)$ w/ $w(t)$ $= \int_{-\infty}^{+\infty} x(t')\, w(t'-t)\, dt'$

$= y(t)$





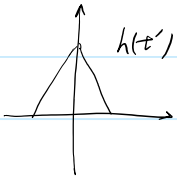
Ise fold then move



$h(t') = w(-t')$

$w(t') = h(-t')$

Symmetrical case



④ Discret Case // in computer.

$x(t) \longrightarrow X[n] \quad (X_n]$

$h(t) \longrightarrow h[n]$

$y(t) \longrightarrow y[n]$

Discrete convolution

$$y[n] = \sum_m X(m)\, h(n-m)$$

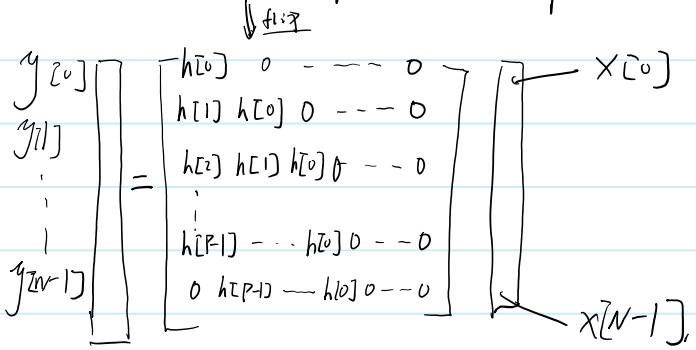
moving ave

$$y[n] = \sum_m x(m)\, w(m-n)$$

2018年3月12日 16:58

$x[0], x[1], \cdots x[N-1]$ input sequence.

$h[0], h[1], \cdots h[P-1] \cdots$ $P << N$



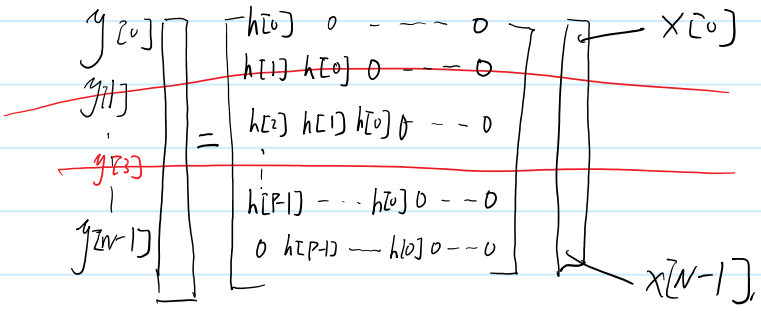
⇓

- matrix very sparse.
- row circular shifted.

⑤ Subsample conv result.

Subsample by 2.

$y[0]$ ✓
$y[1]$
$y[2]$ ✓
$y[3]$
$y[4]$ ✓
⋮



⇓

circular shift by 2

⇒ stride

⑥ Extension to 2D

$$y[m,n] = \sum_{m',n'} x[m',n']\ h[m-m', n-n']$$

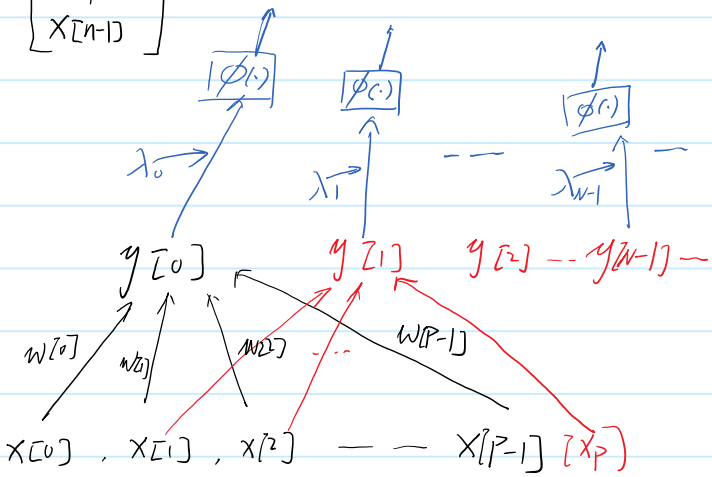
2018年3月12日 16:58

① Neural network implementation of conv. / moving ave.

$$\begin{bmatrix} y[0] \\ y[1] \\ y[2] \\ y[3] \\ y[4] \\ \vdots \end{bmatrix} = \begin{bmatrix} w[0] & w[1] & \cdots & w[P-1] & 0 & \cdots & 0 \\ 0 & w[0] & \cdots & \cdots & w[P-1] & 0 & \cdots \\ 0 & 0 & w[0] & \cdots & \cdots & w[P-1] & \cdots \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[n-1] \end{bmatrix}$$

$w[m] = w[0], w[1], \cdots w[P-1]$

$P << N$.



P weights for entire layer.

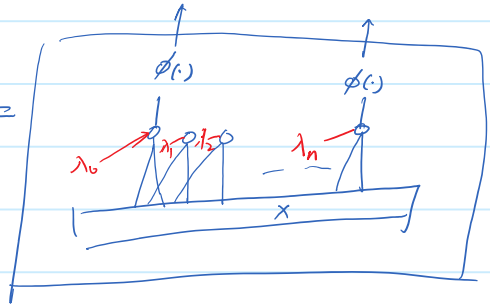
$h = \phi(w * x + \lambda)$

vector — applied component wise (threshold & nonlinearity) — vector — vector

⇓

to have nonlinearity / complexity in function

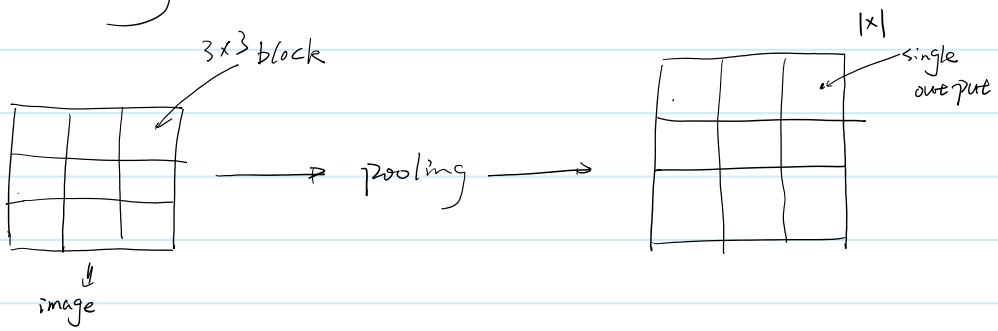
Convolution layer ⇐

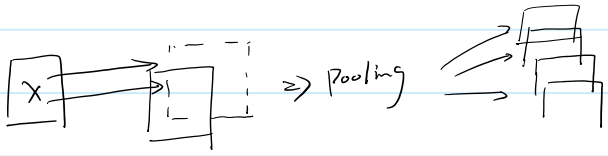


2018年3月12日 16:58

⑧ Pooling

down sampling





feature filtering

2×4 = 8 types

conv layer

(diffirent conv kernel)

ex. diffirent convolutions

⑨ Notable CNNs

LeNet 5

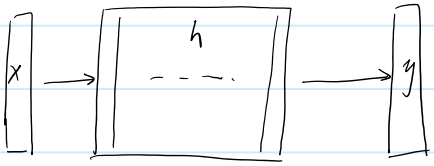
AlexNet

GoogleNet

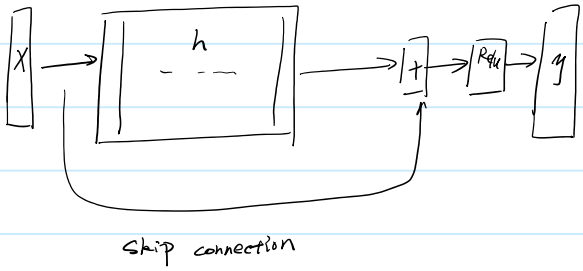
ResNet 152 layers

ResNet

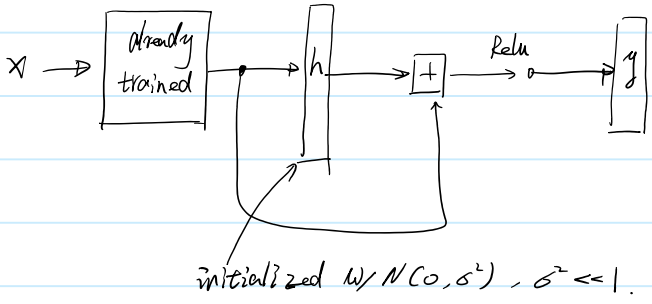
conventional network



resNet



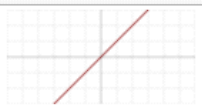
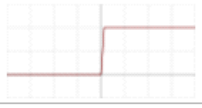
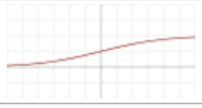
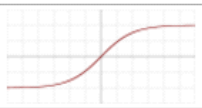
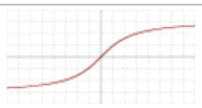
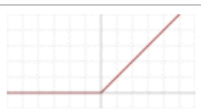
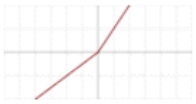
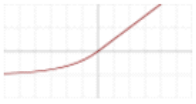
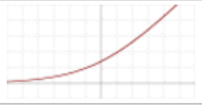
layer-by-layer training



training starting from already trained ; only accept things that makes it better

Sunday, March 25, 2018 10:55 AM

Activation function

Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1-f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2+1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1+e^x)$	$f'(x) = \frac{1}{1+e^{-x}}$

5

2018年3月26日 16:58

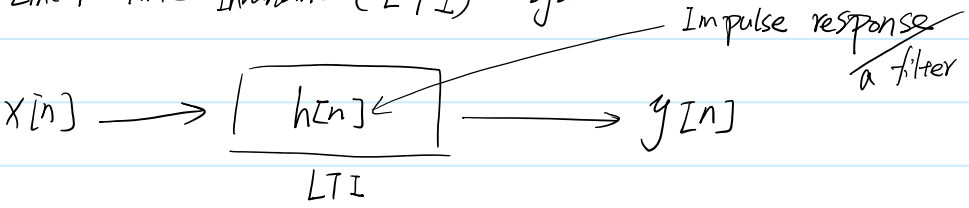
Week9

2018年3月26日 17:38

5. Reccurent Neural Network (RNNs)

1) FIR and IIR filters

① Recall Linear Time Invariant (LTI) system



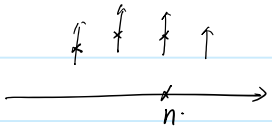
$$y[n] = (h * n)[n]$$

② FIR system (Finite impulse response)

$h[n]$ has finite support

$y[n] = (h*n)[n]$ = local average.

example: $y[n] = \frac{1}{3}(x[n] + x[n-1] + x[n-2])$

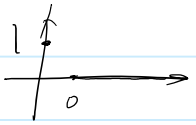


$h[n] = ?$

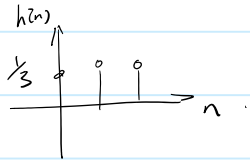
$h[n]$ = impulse response = output (input = $\delta(n)$)

$= \frac{1}{3}[x[n] + x[n-1] + x[n-2]]$ / $x[n] = \delta[n]$

$\delta(n) = \begin{cases} 0, & n \neq 0 \\ 1, & n = 0 \end{cases}$



$= \frac{1}{3}[\delta[n] + \delta[n-1] + \delta[n-2]]$



2018年3月26日 17:39

⑧ IIR (infinite impulse response).

example: $y[n] = a\,y[n-1] + (1-a)\,x[n]$ ← memory

recursive equation

$y[-1] = 0$

$h[n] = ?$ Z transform

$h[n]$ = output (input $x[n] = \delta[n]$)

$h[0] = y[0] = a\,y[0-1] + (1-a)\,x[0] \;\big|\; x[0] = \delta[0]$

($y[0-1]$: 0; $x[0]$: 1)

$= 1-a$

$h[1] = y[1] = a\,y[0] + [1-a]\,x[1] \;\big|\; x[1] = \delta[1]$

($y[0]$: $1-a$; $x[1]$: 0)

$= a(1-a)$

$h[2] = y[2] = a\,y[1] + (1-a)\,x[2] \;\big|\; x[2] = \delta[2]$

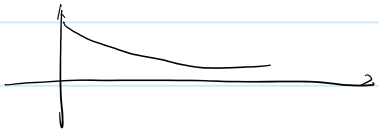
($y[1]$: $a(1-a)$; $x[2]$: 0)

$= a^2(1-a)$

$h[n] = (1-a)\,a^n, \quad n = 0, 1, 2 \cdots$

Support $[h[n]] = +\infty$

$y[n] = [h * x][n]$



$|a| < 1$ for system to be stable

2018年3月26日 17:38

2) Relation to Neural Network

FIR filter ⟶ CNN

IIR filter ⟶ RNN

CNN layer

$x \rightarrow \underset{\text{layer}}{\text{CNN}} \rightarrow y$

$y = \phi((x * h) + \lambda)$

small kernal

$= \phi((W_x x) + \lambda)$

W_x = [ ]

① RNN: Simple RNN

$y[n] = \phi(W_x x[n] + W_y y[n-1] + \lambda)$

vector vector vector vector

non-linear applied component wise

scalar scalar scalar

$y[n] = \phi(a\, y[n-1] + (1-a)\, x[n] + \lambda)$

graph

advantage:

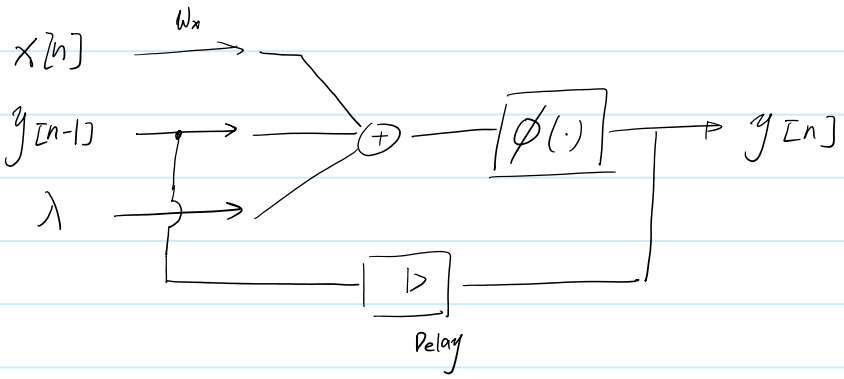
① non-linear

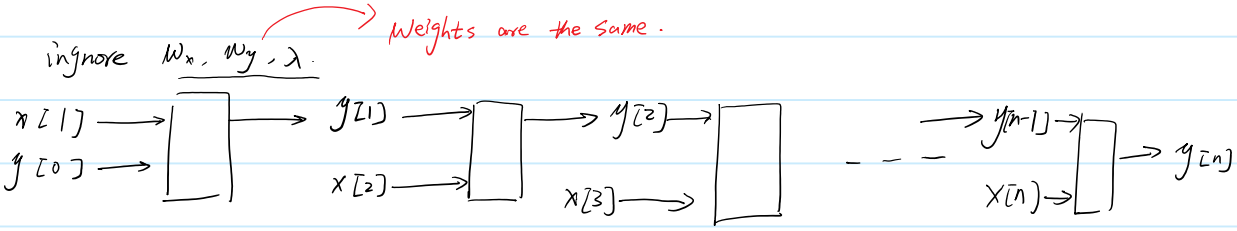
② system with long memory (infinite window/kernal size) — obvious direction in time

③ dimension of input not need to be constant.

2018年3月26日 17:39

② Roll-out representation





Input sequence of length n.

output sequence of length n.

③ Training in RNN

Typically training Data.

Batch

$$D = \{ (x^1[1], y^1[0]), (x^1[2], y^1[1]) \cdots (x^1[n], y^1[n-1]), y^1[n] \}$$

$$(x^2[1], y^2[0]), (x^2[2], y^2[1]) \cdots (x^2[n], y^2[n-1]), y^2[n]$$

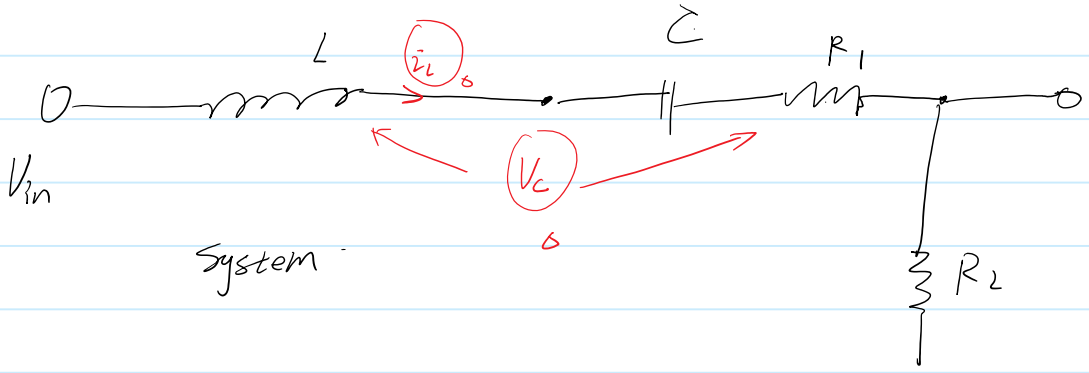
$$\vdots$$

2018年3月26日 17:39

4) More general RNN.

Simple RNN, extension of $y[n] = a\,y[n-1] + (1-a)\,x[n]$

More general RNN, (1) state space representation.



Linear case

State variables $x[\lambda]$

State evolution

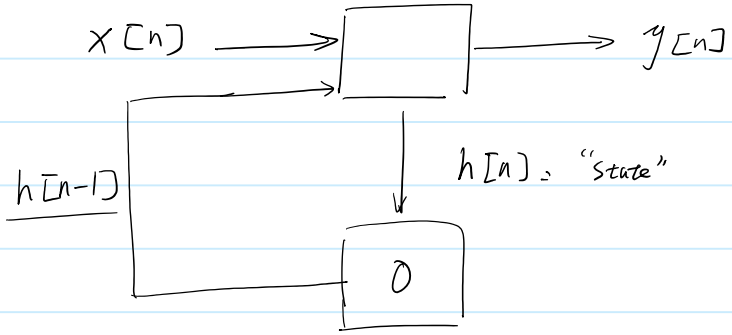
$$x[n+1] = A\,x[n] + w[n]$$

$$y[n] = H\,x[n] + v[n]$$

$y[n]$: observable

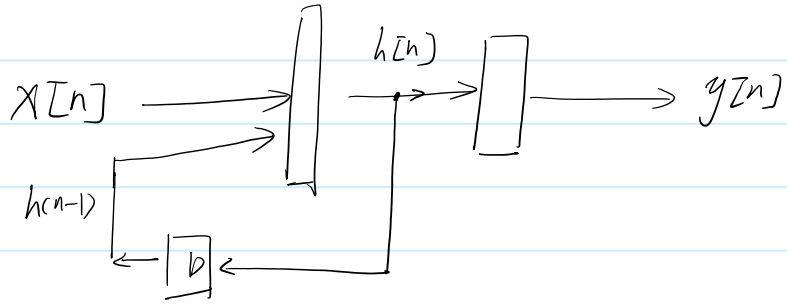
$x[n]$: may or may not be observable

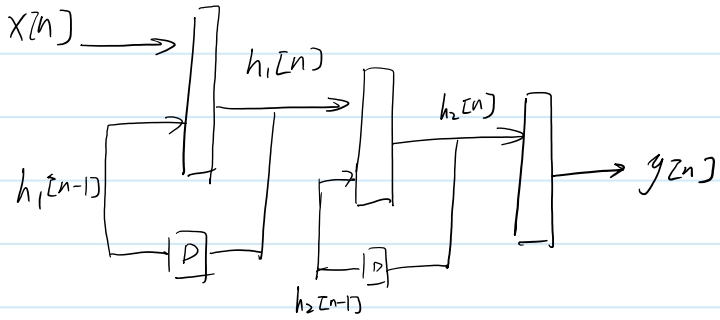
$\Rightarrow E[\,x[n] \mid y[n], y[n-1] \cdots y[0]\,]$



6

2018年3月26日 17:39





Week10

2018年4月2日 17:45

clustering (Grouping)

1) what ?

① Example, cluster/group gradient.

W. HF ? EF.

HF: heart failure.

EF: ejection fraction

< 50% abnormal

?: preserved.

normal

65 features / patient ⟶ x_i

162 patients ⟶ [clustering algorithm] ⟶ k groups

② Data Grouping

w/in a group, similar

btw group, dis-similar

⇒ unsupervised classification

③ key to clustering

— distance measure.

— data model

2.

2018年4月2日 17:44

2) distance measures

I. real valued data.

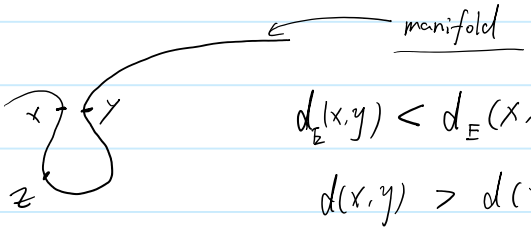
$x \in \mathbb{R}^n$, $y \in \mathbb{R}^n$

$d(x,y) = \|x-y\|$ (Eculidian)

normalized $d(x,y) = \|x-y\|_\Sigma$ ← matrix

$= \sqrt{(x-y)\Sigma^{-1}(x-y)}$

II.



$d_E(x,y) < d_E(x,z)$.

$d(x,y) > d(x,z)$

III. categorical data.

Example.

$x = (yes, yes) \longrightarrow (0,0)$

$y = (no, no) \longrightarrow (1,1)$

$d(x,y) = d_H(x,y) = 2.$

Hamming distance.

categorical but not binary.

Red → 001

Blue → 010

Green → 100

$x = (yes, red) \longrightarrow (0, 001)$

$y = (no, green) \longrightarrow (1, 100)$

$z = (no, blue) \longrightarrow (1, 010)$

$d_H(x,y) = 3.$

3

2018年4月2日 17:45

Problem

$X = (X_1, X_2 \cdots X_n)$.

X_1 = Categorical Variable.

w/ 10^7 categories

one-hot rep of X_1 10^7 long

When does this happen? X_1 is a word.

is a comb. of cate variables.

Word-Embedding entity 10^7 dim. one-hot vector $\longrightarrow$ m dim real vector

- $m \ll 10^7$.
- if x (cat) and y (dog) are similar.

$d(x_1^e, y_1^e)$ small.

Rough Idea: Find the embedding in a classification task.

w/ a neural network

N-gram: $P(w_t / w_{t-1} \cdots w_{t-(N-1)}) = P(w_t / w^{t-1})$

assume to be true prob. distribution for words.

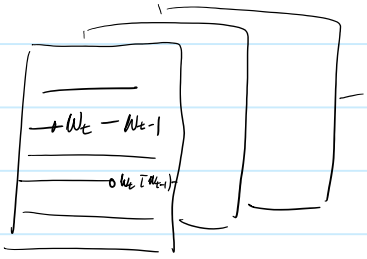
$g(w_t / w_{t-1})$ = estimate of $(P_{w_t / w^{t-1}})$

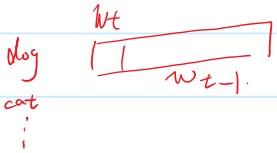
= output a neural network

4 4

2018年4月2日 17:44

Trainning data: document / articles.





$P(\text{dog} \mid w_{t-1}) = 1$

$P(\text{cat} \mid w_{t-1}) = 0$

optimization function: Cross entropy

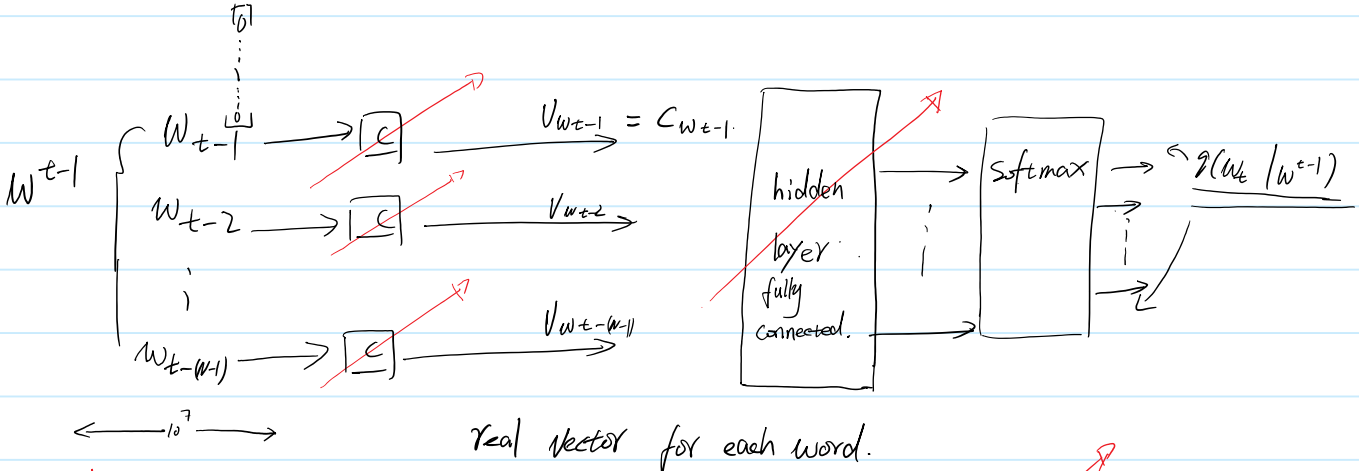
$$\max \sum \; - \sum_{w_{t'}} P(w_{t'} \mid w^{t-1}) \log q(w_{t'} \mid w^{t-1})$$

all possible words

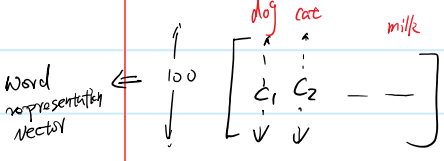
either one or zero.

all training data point.

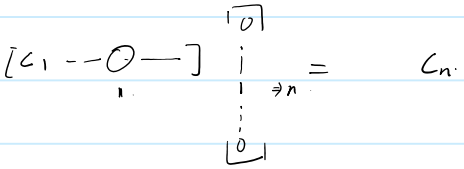
all point in mini-batch.



Real vector for each word.



things change



5

2018年4月2日 17:45

3) Hierachical clustering

(Medical)

bottom up.

Distance – based.

no model.

Adv:

Simple.

works well when there is hierachiy in data

problems.

where to cut

4) K-means LBG, VQ

K: the number of clusters. Given

initialization: start w/ K cluster centers. $C_k^{(0)}$, $k = 1, 2 \cdots, k$.

classification: assign x_i to $C_k^{(n)}$ ← n^{th} iteration $n = 0, 1, 2 \cdots$

$$\text{if } d(x_i, C_k^{(n)}) = \min_{k'} d(x_i, C_{k'}^{n})$$

min distance classification.

Update:

C_k^{n+1} = average of all points in the k^{th} cluster.

Check:

$$\| C^{n+1} - C^n \| < \varepsilon \qquad C = (C_1, C_2, \cdots C_k)^T$$

Yes → stop

No → go to classification

2018年4月2日 17:45

Adv: simple.
works well if
a good K
the data is spherical.

Problems:
K is not known. → solution → modeling
Guassian mixture
1) Info theoretic criteria (AIC, BIC, XIC).
2) non parametric Bayesian.
3) EM Bayesian.

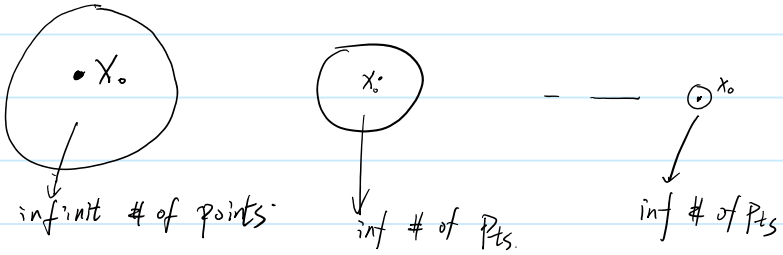
If data is not spherical.

initialization, needs good initialization.

5) Density – Distance clustering

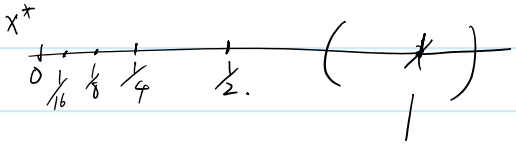
Basic idea. math analysis.

a cluster point:



$D = \{ x_i \mid x_i = 2^{-i} \}$ $i > 0$, integer

cluster point for D: $x^* = 0$



Week11

Monday, April 9, 2018 5:44 PM

Review: clustering Algorithm.

- hierarchical
- kmeans
- density-distance

= } every data point → a cluster.

fuzzy-kmeans

Model-based clustering

- A pdf for data.
- MLE/MAP → cluster

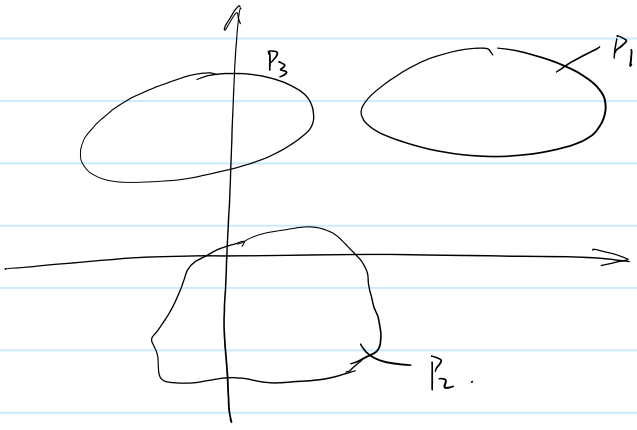
→ EM algorithm

6) Model based clustering & EM Algorithm.

① mixture model

$$P(X) = \sum_{k=1}^{k} \pi_k P_k(x).$$ ← pdfs.

$\pi_k > 0$, $\sum \pi_k = 1$



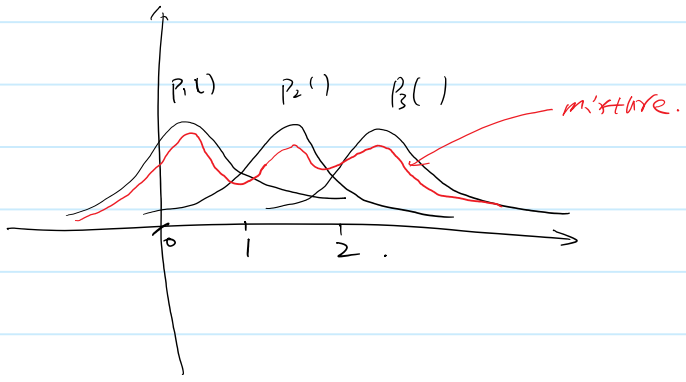
Example. Guassian mixture ← $N(m_k, \sigma_k^2)$

$$P(x) = \sum_{k=1}^{k} \pi_k P_k(x | m_k, \sigma_k^2).$$

e.g. $k=3$. $\pi_1 = 0.2$, $\pi_2 = 0.4$ $\pi_3 = 0.4$

$\sigma_1 = \sigma_2 = \sigma_3 = 1$.

Monday, April 9, 2018 8:38 AM



Relation to clustering

△ fix k.

Data $D=\{y_1, y_2 \cdots y_N\}$ i.i.d.

model: $p(y) = \sum_{k=1}^{k} \pi_k P_k(y)$.

$= \sum_{k=1}^{k} \pi_k P_k(y|\phi_k) = P(y|\theta)$ $\theta = (\{\pi_k\}, \{\theta_k\})$.

MLE
MAP

Training $D \longrightarrow \hat{\theta} \longrightarrow P[y_i \in \text{clustering} \,|\, y_i, \hat{\theta}]$

△ finding k

- info. theoretic criteria.
- NPB (Non-parametric Bayesian).
- ✓ — as a part MLE/MAP

② MLE for clustering

clustering $D \longrightarrow \hat{\theta}$

MLE: $\theta = \arg\max_{\theta} \log P(y_1, y_2 \cdots y_N|\theta) = \arg\max_{\theta} \log P(y|\theta)$

$= \arg\max_{\theta} \log P(D|\theta)$

Monday, April 9, 2018 8:38 AM

$$\log p(D|\theta) = \log p(\overbrace{y_1, y_2, \cdots y_n}^{i.i.d.} | \theta)$$

$$= \sum_{i=1}^{n} \log p(y_i|\theta)$$

$$= \sum \log \sum_{k=1}^{K} \pi_k p_k(y_i|\theta_k) \qquad \theta_k = (m_k, \Sigma_k^2), \quad \theta = (\{\pi_k\}, \{m_k, \Sigma_k\})$$

$$= \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k (2\pi)^{-\frac{\eta}{2}} |\Sigma_k|^{-\frac{1}{2}} \exp\left[-\tfrac{1}{2}\|y_i - m_k\|^2 \Sigma_k^{-1}\right] \right)$$

highly non-linear / non-quadratic

constraint: $\pi > 0, \Sigma \pi_k = 1$

→ difficult to solve

⑧ The EM (expectation — maximization) Algorithm

Basic idea:

Auxiliary variable

$$p(y) = \int p(y|z)\, p(z)\, dz$$

complex — not complex (as)

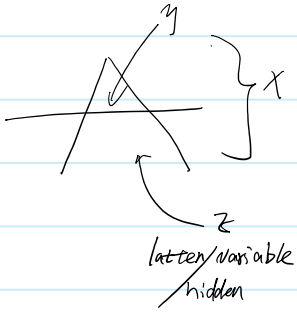
Data $= \{y_1, y_2, y_3 \cdots y_N\} = y$

→ Topics Model

Introduce x: complete data.

y: incomplete data.

$y = H(x)$



— y incomplete data

— y, z complete data $= x$.

$(1, 0)\begin{pmatrix} y \\ z \end{pmatrix} = y$

H

Monday, April 9, 2018 8:38 AM

MLE: $\hat{\theta} = \arg\max_{\theta} \log(y|\theta)$ Difficult if done directly.

EM: iterative btw E and M steps.

Start: initial guess — $\theta^{(0)}$

E-step: generate function

$$\Phi(\theta|\theta^{(p)}) = E[\log p(x|\theta) | y, \theta^{(p)}]$$

($\theta^{(p)}$: current estimate)

$(\approx \log p(y|\theta))$

good current appr. to $\log p(y|\theta)$

M-step:

$$\theta^{(p+1)} = \arg\max_{\theta} \Phi(\theta|\theta^{(p)})$$

convergence: $\log p(y|\theta^{(p+1)}) \geq \log p(y|\theta^{(p)})$

convergence to local max.

Can be highly dependent on initial guess.

Monday, April 9, 2018 8:39 AM

⑥ Applied to clustering

— Find/Def compelete data.

— E-step can be difficult.

— M-step easy

Incompelete data $y = \{y_1, y_2, \cdots y_N\}$ (i.i.d)

$$P(y) = \sum \pi_k P_k(y)$$ (Gaussian)

compelete data: $x = \{y, z\}$

$z = \{z_1, z_2, \cdots z_N\}$ = 'indicator vectors'

$z_i = k$, if $y \in$ cluster k.

z_i: binary indicator vector

$= [0, 0, 0, 1, \cdots 0]$ If $y_i \in$ cluster k.

↓ k.

= 'one-hot' vector

$$y = (\underbrace{I}_{\text{Identity}}, 0)\begin{pmatrix} y \\ z \end{pmatrix}$$

E-step.

$Q(\theta/\theta^{(t)})$

$$\log p(x/\theta) = \log p(y, z/\theta)$$

$$= \log p(y/z) + \log p(z)$$

$$= \sum \log p(y_i/z_i) + \sum \log p(z_i).$$

$$= -\sum \log p(y_i/z_i) + V_\pi^T \sum z_i$$

$\log p(z_i)$

$p(z_i) = \begin{cases} \pi_1, z_i = 1 = e_1 = [1, 0, 0, \cdots 0] \\ \pi_2, z_i = 2 = e_2 = [0, 1, 0, \cdots 0] \\ \vdots \\ \pi_k, k = e_k = [0, 0, 0 \cdots 1] \end{cases}$

vector z_i

$= [\log \pi_1, \log \pi_2 \cdots \log \pi_k]\, z_i$

$= V(\pi)^T z_i \qquad \pi = (\pi_1, \pi_2 \cdots \pi_k)$

$V_\pi = \begin{bmatrix} \log \pi_1 \\ \vdots \\ \log \pi_k \end{bmatrix}$

Monday, April 9, 2018 8:39 AM

$$\log p(y_i | z_i) = [\log p(y_i|e_1), \cdots \log p(y_i|e_k)]\, z_i$$

$$= u^T(y_i)\, z_i$$

$$u^T(y_i) = [\log p(y_i|k_1)_{m_1,\Sigma_1}, \ldots \log p(y_i|e_k]_{m_k,\Sigma_k}$$

$$= u^T(y_i, m, \Sigma)$$

$$m = (m_1, m_2, \ldots m_k)$$

$$\Sigma = (\Sigma_1 \; - \; - \; \Sigma_k)$$

$$\Rightarrow \sum u^T(y_i, m, \Sigma) z_i + V^T(\pi) \sum z_i .$$

$$\phi(\theta|\theta^{(p)})$$

$$= E[\log p(x|\theta) \,|\, y, \theta^{(p)}]$$

$$= E[\sum u^T(y_i, m, \Sigma) z_i + V(\pi) z_i \,|\, y, \theta^{(p)}]$$

$$E[z_i | y, \theta^{(p)}]$$

$$z_i = \begin{bmatrix} z_{i1} \\ z_{i2} \\ \vdots \\ z_{ik} \end{bmatrix} \Rightarrow E[z_{ik} | y, \theta^{(p)}] = P[z_{ik} = 1 \,|\, y, \theta^{(p)}]$$

$$= P[y_i \in \text{cluster } k \,|\, y_i, \theta^{(p)}]$$

$$= P[z_i = e_k \,|\, y_i, \theta^{(p)}]$$

x	0	1
P	P	(1-P)

$E[X] = 0 \cdot P + 1 \times (1-P) = P[X=1]$

$$E[z_{ik} | y, \theta^{(p)}] = P[z_i = e_k \,|\, y_i, \theta^{(p)}]$$

$$\frac{P[y_i | z_i = e_k, \theta^{(p)}] \cdot P[z_i = e_k | \theta^{(p)}]}{\sum_{k'=1}^{k} P(y_i | z_i = e_{k'} | \theta^{(p)}) \cdot P(z_i = e_{k'} | \theta^{(p)})}$$

2018年4月9日 19:36

$$= \frac{P_k(y_i \mid m_k^{(p)}, \Sigma_k^{(p)}) \cdot \pi_k^{(p)}}{\sum_{k'=1}^{K} P_{k'}(y_i \mid m_{k'}^{(p)}, \Sigma_{k'}^{(p)})}$$

$$= E[z_{ik} \mid y, \theta^{(p)}] = \langle z_{ik}^{(p)} \rangle = \bar{z}_{ik}^{(p)} = P[y_i \in k^{th} \text{ cluster} \mid \text{current parameters est.}]$$

$$\bar{Q}(\theta \mid \theta^{(p)}) = \sum U^T(y_i, m\Sigma)\, \bar{z}_i^{(p)} + V(\pi) \sum \bar{z}_i^{(p)}$$

$$= \sum U_i^T \bar{z}_i^{(p)} + V^T \bar{z}_i^{(p)}$$

M-step, $\theta^{(p+1)} = \arg\max_{\theta} \bar{Q}(\theta \mid \theta^{(p)})$

$$\nabla_\theta \bar{Q}(\theta \mid \theta^{(p)}) = 0$$

$$\sum (\nabla_\theta U_i^T)\, \bar{z}_i^{(p)} + \left(\nabla_\theta V^T(\pi)\right) \mathcal{E}\, \bar{z}_i^{(p)} = 0$$

$\pi_k > 0 \quad \sum \pi_k = 1.$

Probability

$$m_k^{(p+1)} = \frac{\sum \bar{z}_{ik}^{(p)} \cdot y_i}{\sum \bar{z}_{ik}^{(p)}}$$

$\sum \bar{z}_{ik}^{(p)}$ → # of points in cluster k.

$$\Sigma_k^{(p+1)} = \frac{\sum \bar{z}_{ik}^{(p)} (y_i - m_k^{(p+1)})(y_i - m_k^{(p+1)})^T}{\sum \bar{z}_{ik}^{(p)}}$$

$$\pi_k^{(p+1)} = \frac{\sum \bar{z}_{ik}^{(p)}}{N}$$

8

2018年4月9日 19:36

7) Determine the # of clusters for k-means and EM based model

① Information Theoratic Criteria.

AIC
Bayesian. BIC
⋮
XIC

$\Rightarrow \log P(y|\theta) + \text{Penalty}(\theta)$

Penalty (θ) = complexity of θ
complexity of model.

② NPB - Non-parametric Bayesian.

Parametric $P(y) = P(y|\theta)$ (θ: parameter)

$\hat{\theta} = \arg\max_{\theta} \log P(y|\theta)$

Non-parametric $P(y) = P(y)$

$\hat{P} = \arg\max_{P} \log P(y)$

$\hat{P} = \arg\max_{P} \{ \log P(y) + \log \text{Prior}(P) \}$

③ MAP EM

$\hat{\theta} = \arg\max_{\theta} \{ \log P(y(\theta)) + \log P(\theta) \}$

Week12

2018年4月16日 17:37

Clustering

Review: EM algorithm

y = observation data $\quad y = D$

MLE, $\hat{\theta} = \arg\max_{\theta} \log P(y|\theta)$ ← $\pi P(y|\theta)$

$= \pi \sum \pi_k P_k(y|\theta_k)$

$= \arg\max_{\theta} \log P(D|\theta)$

EM algorithm.

λ (e.g. $X = (y, z)$) ← latent hidden variables

E-step $\quad \phi(\theta|\theta^{(t)}) = E[\log P(X|\theta) | y, \theta^{(t)}]$

(Calculating: $E[z|y, \theta^{(t)}]$)

M-step $\quad \theta^{(t+1)} = \arg\max_{\theta} \phi(\theta|\theta^{t})$

Determine K (# of clusters)

MAP EM

2018年4月16日 17:37

MAP EM.

Prior on θ $P(\theta)$

E-step: same as EM.

M-step: $\theta^{(P+1)} = \arg\max_{\theta} \{ \Phi(\theta | \theta^{(P)}) + \log P(\theta) \}$

Possible prior for θ

final clusters equations for MAP EM

---

3 clusters 3 labels

Average purity

④ priors for mixture model parameters

mixture model:

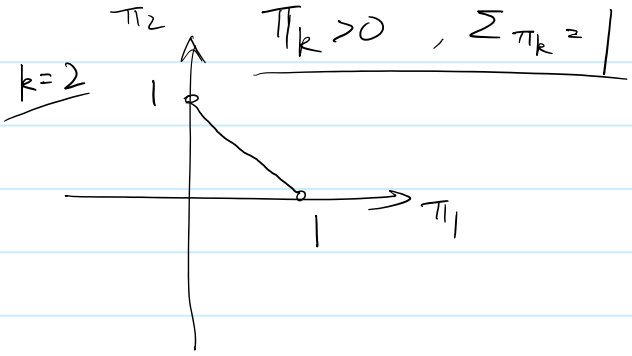
$$P(y) = \sum_{k=1}^{K} \pi_k P_k(y | m_k, \Sigma_k)$$

parameters $\theta = \{ \pi_k, (m_k, \Sigma_k) \}$ ← joint prior for $\{m_k, \Sigma_k\}$

prior for $\{\pi_k\}$

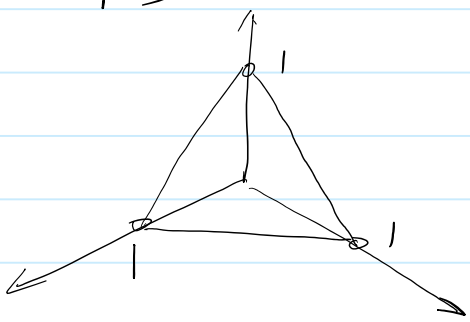
2018年4月16日 17:37

⑤ Prior for $\{\pi_k\}$



$\pi_k > 0$, $\sum \pi_k = 1$

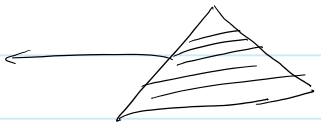
$k=3$



$\alpha_1 = \alpha_2 = \alpha_3 = 1$

Dirichlet Distribution

uniform Density



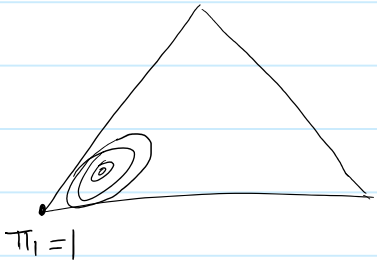
$\pi = \{\pi_1, \cdots \pi_k\}$ $\alpha = (\alpha_1, \cdots \alpha_k) > 0$

$$P(\pi|\alpha) = \frac{1}{B(\alpha)} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}$$

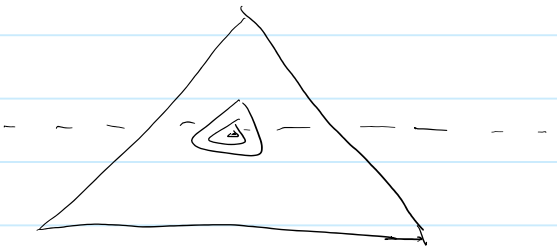
If $\alpha_k = 1$, all $k \Rightarrow$ uniform distribution

Ref: Bishop

If $K=3$, $\alpha_1 = 20$, $\alpha_2 = 2$, $\alpha_3 = 2$.



If $\alpha_1 = \alpha_2 = \alpha_3 = 2$



4

2018年4月16日 17:37

⑤ Prior for m_k, Σ_k

→ NIW distribution

Normal inverse Wishart

Simpler case: prior for m_k.

Prior for m.

likelihood / Data model: Gaussian $p(y|m,\Sigma)$ — mean, cov. matrix

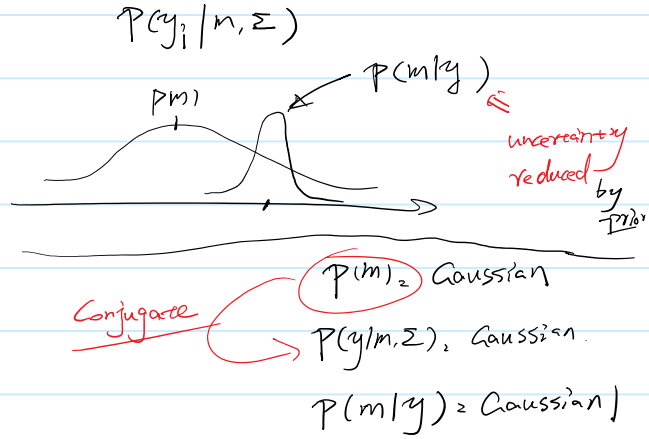
Prior on m: $m \sim N(m_0, \Sigma_0)$ — constant hyper parameters

Posterior $p(m|y)$ $\{y_1, y_2 \cdots y_N\}$

Bayesian inference

$p(m) \longrightarrow p(m|y)$

Prior



Special case: y: 1 dim.

$p(y|m,\sigma) = p(y|m,\sigma^2)$

$p(m) = p(m|m_0, \sigma_0^2)$

$p(m|y) = p(m|y_1 \cdots y_N)$

$= p(m|m_N, \sigma_N^2)$

2018年4月16日 17:37

Special case y : 1 dim.

$$p(y|m,z) = p(y|m,\sigma^2)$$

$$p(m) = p(m|m_0,\sigma_0^2)$$

$$p(m|y) = p(m|y_1,\cdots y_N)$$

$$= p(m|m_N,\sigma_N^2)$$

$$m_N = \frac{\frac{N}{\sigma^2}}{\frac{N}{\sigma^2}+\frac{1}{\sigma^2}}\left(\frac{\sum y_i}{N}\right) + \frac{\frac{1}{\sigma_0^2}}{\frac{N}{\sigma^2}+\frac{1}{\sigma_0^2}} m_0$$

$\left(\frac{\sum y_i}{N}\right) = \hat{m}_{mL}$

$$\frac{1}{\sigma_N^2} = \frac{1}{\sigma_0^2} + \frac{N}{\sigma^2} \longrightarrow \sigma_N^2 = \frac{1}{\frac{1}{\sigma_0^2}+\frac{N}{\sigma_0^2}}$$

When $N \to +\infty$

$m_N \to \hat{m}_{mL}$

$\frac{1}{\sigma_N^2} \to \frac{\sigma^2}{N} \to 0$

---

n-dim case

$$p(y|m,\Sigma)$$

$$p(m|m_0,\Sigma_0)$$

$$p(m|y) = p(m|m_N,\Sigma_N)$$

$$\Sigma_N^{-1} = \Sigma_0^{-1} + N\Sigma^{-1}$$

$$m_N = \Sigma_N^{-1}(\Sigma^{-1} N \hat{m}_{mL} + \Sigma_0^{-1} m_0)$$

2018年4月16日 19:12

Prior for Σ

$P(y|m,\Sigma)$: Guassian

↑ fixed.

Wishart inverse distribution.

$P(\Sigma|S_0,\nu_0)$: $IW(\Sigma|S_0,\nu_0)$

$= c|\Sigma|^{-(\nu_0+D+1)/2} \cdot \exp\left(-\frac{1}{2}tr(S_0\Sigma^{-1})\right)$

prior matrix for Σ

trace

$D = \dim(y)$

$= m$.

$E[\Sigma] = \frac{S_0}{\nu_0 - D - 1}$

$(\nu_0 > D+1)$

ν_0 – Degree of freedom

$A = [a_{ij}]$ (square)

$tr[A] = \sum a_{ii}$

Property: $trace(ABC) = trace(CAB)$

$\Rightarrow trace(X^T\Sigma^{-1}X)$

$= trace(XX^T\Sigma^{-1})$

A number ⟸

Posteriors for Σ.

$P(\Sigma|m,y) = IW(\Sigma|S_N,\nu_N)$

$P(y|m,\Sigma)$: Guassian.

$S_N = S_0 + S_m$ $\quad S_m = \sum^{sum}(y_i-m)(y_i-m)^T$

$\nu_N = \nu_0 + N$

7

2018年4月16日 18:45

Prior mean

$$E[\Sigma] = \frac{S_0}{\nu_0 - D - 1}$$

Posterior mean

$$E[\Sigma | m, y] = \int IW(\Sigma | S_N, \nu_N) \cdot \Sigma \, d\Sigma$$

$$= \frac{S_N}{\nu_N - D - 1} = \frac{S_0 + S_m}{\nu_0 + N - D - 1}$$

($S_m = \Sigma (y_i - m)(y_i - m)^T$)

$$= \frac{1}{\nu_0 + N - D - 1} S_0 + \frac{N}{\nu_0 + N - D - 1} \cdot \frac{S_m}{N}$$

$\frac{S_m}{N}$: MLE of Σ

When $N \to +\infty$

$\Rightarrow$ MLE of Σ

Prior $p(m, \Sigma)$

if $p(m, \Sigma) = p(m) \cdot p(\Sigma)$

$\longrightarrow$ not conjugate wrt $p(y | m, \Sigma)$

Conjugate prior

$$p(m, \Sigma) = p(m | \Sigma) \cdot p(\Sigma)$$

$$= p(m | m_0, \frac{1}{k_0}\Sigma) \cdot IW(\Sigma | S_0, \nu_0) = NIW(m, \Sigma | m_0, k_0, S_0, \nu_0)$$

(Σ: variable.)

Posteriors → conditional mean. / mode (MAP estimation)

$\longrightarrow$ conjugate to Gaussian likelihood.

$\Rightarrow$ Posterior $p(m, \Sigma | y) = NIW(m, \Sigma | m_N, k_N, S_N, \nu_N)$

$$m_N = \frac{k_0}{k_0 + N} m_0 + \frac{N}{k_0 + N} \left(\frac{\Sigma y_i}{N} \right)$$

$$k_N = k_0 + N$$

$$\nu_N = \nu_0 + N$$

($S = \Sigma y_i y_i^T$)

$$S_N = S_0 + S + k_0 m_0 m_0^T + k_N m_N m_N^T$$

2018年4月16日 18:46

MAP estimate.

$$= \underset{m,\Sigma}{\operatorname{argmax}}\ p(m, \Sigma \mid y)$$

$$= \left(m_N, \frac{S_N}{\nu_N + D + 2}\right)$$

---

① Applications to clustering (i.e. mixture model)

$\Rightarrow$ MAP EM.

Recall, $D = \{y_1, y_2, \cdots y_N\} = y$.

$$p(y) = \sum \pi_k p(y \mid m_k, \Sigma_k)$$

$$= p(y \mid \theta)$$

$$\theta = \{\pi_k, m_k, \Sigma_k\}$$

Complete data. $x = (y, z)$

z = hidden variables $= \{z_1, z_2 \cdots z_N\}$

$z_i = e_k$ if $y_i \in k^{th}$ cluster.

↑ 1 hot vector

$$= \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_k \end{bmatrix}$$

2018年4月16日 18:46

E-step : $Q(\theta|\theta^{(p)})$

$$= E[\log p(x|\theta) | y, \theta^{(p)}]$$

$$= \sum_i \sum_k \left(\bar{z}_{ik} \log \pi_k + \bar{z}_{ik} \log p(y_i | m_k, \Sigma_k) \right)$$

$\bar{z}_{ik}$: $E[z_{ik} | y, \theta^{(p)}]$

M-step (MAP : M-step)

$$\theta^{(p+1)} = \arg\max \{ Q(\theta|\theta^{(p)}) + \log p(\theta) \}$$

$$\log p(\theta) = \log p(\pi) + \sum \log p(m_k, \Sigma_k)$$

$p(\pi)$: $\{\pi_1, \pi_2 \cdots \pi_k\}$ Dirichlet

$p(m_k, \Sigma_k)$: $NIW(m_k, \Sigma_k | m_0, k_0, v_0, S_0)$

$$m_k^{(p+1)} = \frac{\bar{z}_k}{\bar{z}_k + k_0} m_{k_{ML}} + \frac{k_0}{\bar{z}_k + k_0} m_0$$

$$\bar{z}_k = \sum_{k=1}^{N} \bar{z}_{ik}$$

$m_{k_{ML}}$: $\frac{\sum_i z_{ik} y_i}{\sum_i z_{ik}}$

$$\Sigma_k = \frac{S_0 + S_k}{v_0 + \bar{z}_k + D + 2} + \frac{k_0 \bar{z}_k}{k_0 + \bar{z}_k} (m_{k_{ML}} - m_0)(m_{k_{ML}} - m_0)^T$$

$$S_k = \sum_{\text{sum}} (y_i - m_{k_{ML}})(y_i - m_{k_{ML}})^T$$

$$\pi_k = \frac{\bar{z}_k + \alpha_k - 1}{N + \sum_k \alpha_k - K}$$

Week13

2018年4月16日 18:46

7. Variational Influence.

observation

$\log P(y|\theta)$

1) Why?

Influence:

→ MLE: $\hat{\theta} = \arg\max_{\theta} \log P(D|\theta)$

→ MAP $\hat{\theta} = \arg\max_{\theta} \{\log P(D|\theta) + \log P(\theta)\}$

→ Posteria $P(z|y,\theta)$

hidden variable

observed variable.

Problem:

analytic derivation can be difficult.

→ influence difficult.

Solution:

EM algorithm. (require $P(z|y,\theta)$)

$P(z|y,\theta^{(t)})$

can be calculated for mixture case.

However, many cases can't be calculated.

Solution: Variational influence.

2018年4月23日 17:02

Basic idea of Variational Influence.

p: difficult to work with.

$\Rightarrow$ find q, good appr. to p.

work with q.

① How to measure the difference between p & q.

② How to find the closest q for p?

2) Measure of difference of two pdfs (continous) / prob. distribution (discrete)

$p = p(x)$, pdf.

$q = q(x)$, pdf.

$$d(p,q) = SQE = c\int_{-\infty}^{+\infty} (p-q)^2(x)\,dx$$

$(p(x)-q(x))^2$

also, $c\int_{-\infty}^{+\infty} (p-q)^2 p(x)\,dx$ or $q(x)$

KL

Not distance

Directed

KL divergence

ave code length for data comes from p but coded as if it comes q $\longrightarrow$ $H(p,q) - H(q)$ $\longleftarrow$ ave code length for data with p

$$D(p//q) = \int_{-\infty}^{+\infty} p(x)\log\frac{p(x)}{q(x)}\,dx = \int_{-\infty}^{+\infty} p(x)\left(\log p(x) - \log q(x)\right)dx \geq 0$$

$$D(q//p) = \int_{-\infty}^{+\infty} q(x)\log\frac{q(x)}{p(x)}\,dx$$

3

2018年4月23日 17:02

3) Variational (Caculus of variation)

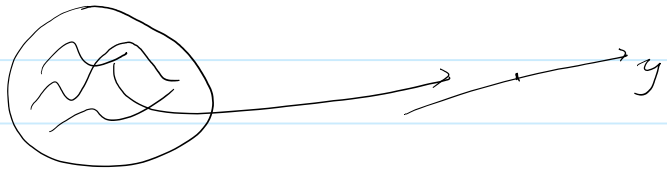
function: $y = f(x)$ (e.g. $y = f(x) = x^2$)

mapping $\mathbb{R} \Rightarrow \mathbb{R}$

△ functional: $F(g) = y$

function → g; number → y

mapping: Function space $\longrightarrow \mathbb{R}$



$$F(g) = \int_{-\infty}^{+\infty} |g(x)|^2 dx$$

$g = g(x)$

Operator:

mapping function $\longrightarrow$ function.

$$H(g) = (g * h)(x)$$

4

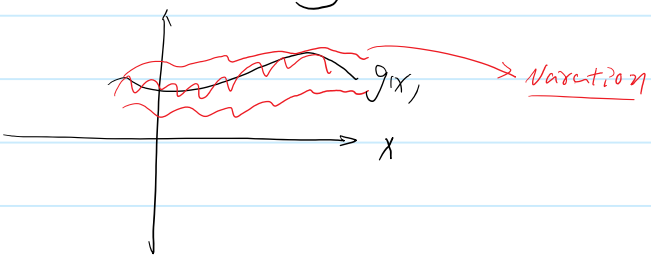
2018年4月23日 17:02

Optimize a function : $\frac{df}{dx} = 0$ (e.g. $f(x) = x^2$, $\frac{df}{dx} = 2x = 0 \Rightarrow x = 0$)

optimize a functional : $\frac{\delta F}{\delta g}$

Small variation in F

Small variation in g



Special case

$\delta g(x) = g_\theta(x)$ (e.g. $g(x) = g_\theta(x)$

$\frac{\delta F}{\delta g} = \nabla_\theta F$ $= N(x / \underbrace{m, \sigma^2}_{\theta})$

Example for special case,

X, random variable

Y, random variable.

$\hat{Y} = g(x)$ = function estimation of Y

$= g_{a,b}(x) = ax + b$

$g^* = \arg\min_g E[(Y - g(x))^2]$ $\Rightarrow$ min a functional.

add constrains $\rightarrow$ $= \arg\min_{a,b} E[\underbrace{(Y - (ax+b))^2}_{\varepsilon}]$ $\frac{\partial \varepsilon}{\partial a} = 0$, $\frac{\partial \varepsilon}{\partial b} = 0$

2018年4月23日 17:02

A more general case.

$$F(g) = \int L(x, g(x), g'(x))\, dx$$

$$\frac{\delta F}{\delta g} = 0 \qquad \frac{\partial L}{\partial g} - \frac{d}{dx}\frac{\partial L}{\partial g'} = 0$$

$$\Rightarrow g(x)$$

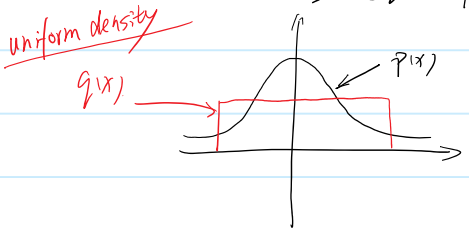
Example, X, random variable.

$p(x) \sim N(0, \sigma^2)$ → fixed

$q(x) \sim U[-a, +a]$, $a > 0$ → various

find $q(x)$ such that

$D(q \| p)$ is minimized.



6

2018年4月23日 17:02

$$D(q\|p) = \int_{-\infty}^{+\infty} q(x) \log \frac{q(x)}{p(x)} dx$$

c $\quad c \cdot 2a = 1$

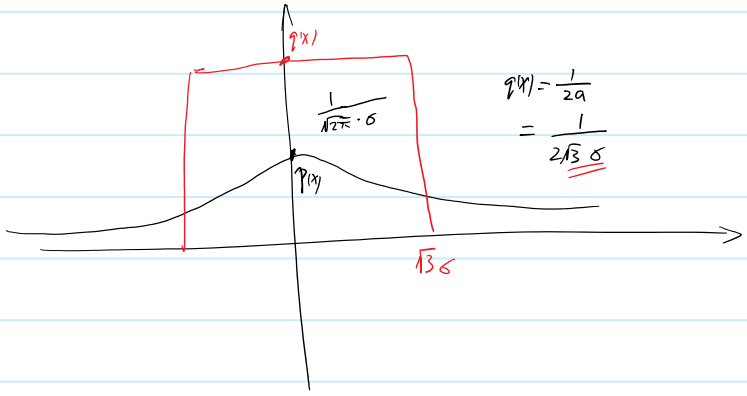
$$= \int_{-a}^{+a} \frac{1}{2a} \cdot \log \frac{\frac{1}{2a}}{\frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{x^2}{3\sigma^2}}} dx.$$

$$= c - \log a + \frac{a^2}{6\sigma^2}$$

$$\frac{d}{da} = 0 \Rightarrow -\frac{1}{a} + \frac{a}{3\sigma^2} = 0$$

$$\Rightarrow a^2 = 3\sigma^2$$

$$a = \sqrt{3}\sigma$$



In practice

$P(\,)$: real distribution

$q(\,)$: approximation, easier to work with

△ $q = \arg\min_q D(p\|q)$, $q = \arg\min_q D(q\|p) \Rightarrow E[q \cdot \log \frac{q}{p}]$

$E(p \cdot \log \frac{p}{q})$ → Better approximation, hard to optimize

→ easier to optimize.

2018年4月23日 17:02

4). Variational / lower bound

Application: Interested in a posterior.

x — data, observable.

z — hidden / latent variable.

$MLE = \theta^* = \arg\max_\theta \log p(x|\theta)$ ← hard to work with.

Posterior: $p(z|x,\theta)$ ← hard to derive or

Week14

2018年4月30日 18:16

2

2018年4月30日 13:37

3

2018年4月30日 13:38

4

2018年4月30日 13:37

5

2018年4月30日 13:38

6

2018年4月30日 13:37

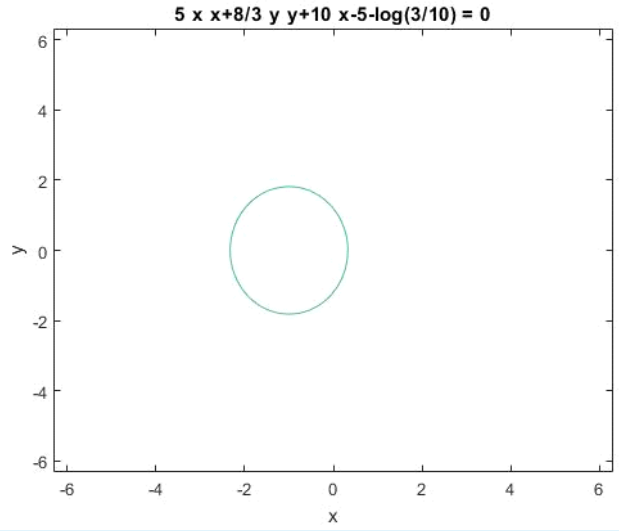
7

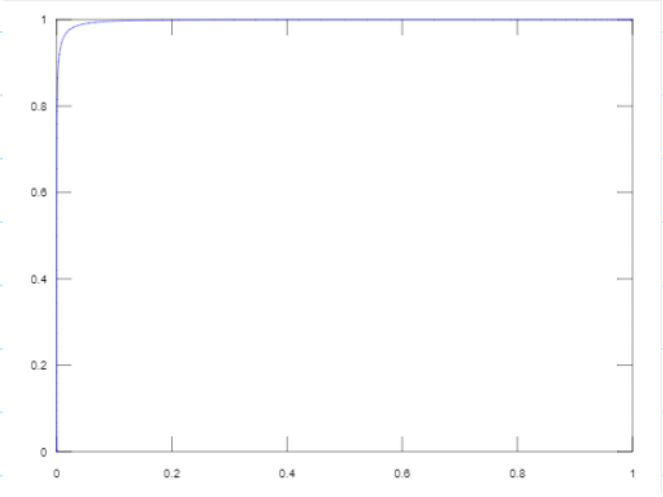
2018年4月30日 13:38

8

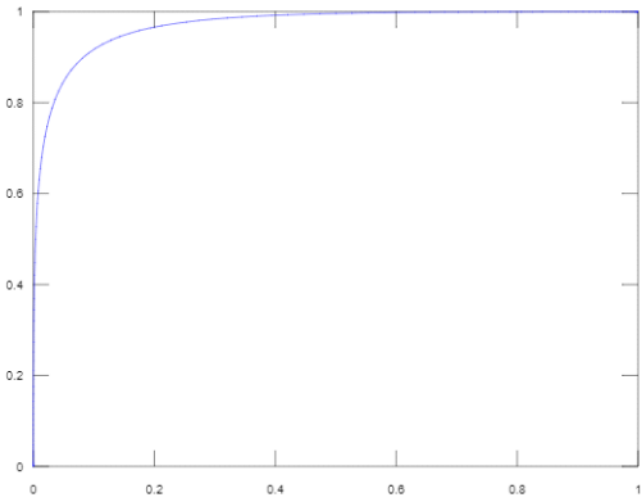
2018年4月30日 13:38

Appendix

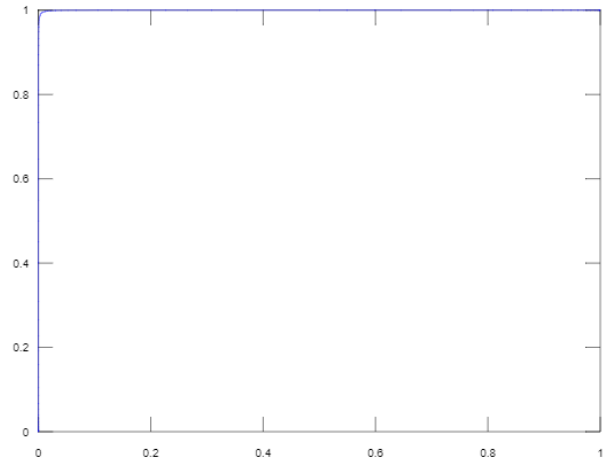




Sigma=1



Sigma=1.5



Sigma=0.8